Model Driven Robotic Assistance for Human-Robot Collaboration

by

Tian Xia

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

October, 2013

Abstract

While robots routinely perform complex assembly tasks in highly structured factory environments, it is challenging to apply completely autonomous robotic systems in less structured manipulation tasks, such as surgery and machine assembly/repair, due to the limitations of machine intelligence, sensor data interpretation and environment modeling. A practical, yet effective approach to accomplish these tasks is through human-robot collaboration, in which the human operator and the robot form a partnership and complement each other in performing a complex task. We recognize that humans excel at determining task goals and recognizing constraints, if given sufficient feedback about the interaction between the tool (e.g., end-effector of the robot) and the environment. Robots are precise, unaffected by fatigue and able to work in environments not suitable for humans.

We hypothesize that by providing the operator with adequate information about the task, through visual and force (haptic) feedback, the operator can: (1) define the task model, in terms of task goals and virtual fixture constraints through an interactive, or immersive augmented reality interface, and (2) have the robot actively

ABSTRACT

assist the operator to enhance the execution time, quality and precision of the tasks. We validate our approaches through the implementations of both cooperative (i.e., hands-on) control and telerobotic systems, for image-guided robotic neurosurgery and telerobotic manipulation tasks for satellite servicing under significant time delay.

Primary Reader:

Peter Kazanzides, Associate Research Professor, Department of Computer Science

Readers:

Russell Taylor, Professor, Department of Computer Science

Louis Whitcomb, Professor, Department of Mechanical Engineering

Acknowledgments

First and foremost, I would like to thank my Ph.D advisor, Dr. Peter Kazanzides. He has consistently offered insights to difficult problems I encountered in my research and helped me to clarify the thesis research direction. Whenever I get off track, he would patiently offer words of encouragement to help me get work done. His tireless work ethnic has made a great impression on me. It truly has been a great pleasure to work with Dr. Peter Kazanzides.

I wish to thank Dr. Russell Taylor for serving on my committee as a reader and for serving on my Graduate Board Oral (GBO) exam. Despite his busy schedule, he has always generously offered his time and insightful comments whenever I approached him for research advice. I am also thankful for Dr. Louis Whitcomb for serving on my committee as reader and for serving on my GBO exam. His "Introduction to Robotics" class was one of the best classes I took in graduate school; his clarity of explanation gave us intuitive understanding of often abstract subjects. I will always remember this quote from Dr. Whitcomb, "All simulations are doomed to succeed"; it carried me through my frustrations of developing complex robotic systems.

ACKNOWLEDGMENTS

I wish to thank Dr. Simon Leonard contributing to the discussions of this dissertation, and the implementation and experiments necessary to complete this dissertation. He is very knowledgeable in the practical implementation of robotic systems, and offered hands-on help to fix seemingly intractable bugs and problems we encountered. I wish to thank Dr. Ankur Kapoor, whose thesis work provided the theoretical foundations and implementations for the the various robotic components used in this dissertation.

I must thank other faculties, staff and students in the Laboratory for Computational Sensing and Robotics at the Johns Hopkins University for their helpful advices and assistance: Dr. Iulian Iordachita, Dr. Greg Hager, Dr. Darius Burschka, Dr. Noah Cowan, Dr. Allison Okamura, Dr. Andreas Terzis, Elisa Ahmanson, Alison Morrow, Min Yang Jung, Marcin Balicki, Tamas Haidegger, Anton Deguet, Balazs Vagvolgyi, Carol Reiley, Dan Mirota, Tom Wedlick, Eatai Roth, John Swensen, Tricia Gibo, Wen Liu, Tomonori Yamamoto, Lawton Verner, Ioana Fleming, Henry Lin, Zihan Chen, Isha Kandaswamy, as well as Cathy Thornton and Debbie DeFord at the Computer Science Department office. I would like to thank my friends, Dan Schlattman, Suzanne Schlattman, Vi Nguyen, Harmonie Sahalov, Rongke Lin, Yuanfan Zhang for making my graduate school life so enjoyable.

I also would like thank my fiance Peng Wu, for her patience, love and encouragement. I am grateful to my parents, Dr. Guanguang Xia and Mintai Liu, for their sacrifices, words of wisdom and both positive criticism and encouragement, without

reservation or exception.

Dedication

To my parents, for all their sacrifices, words of wisdom, and encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

While robots routinely perform dexterous tasks to assemble electronics and cars in structured factory environments, robots are currently unable to autonomously perform certain tasks such as surgery, due to the difficulties in high-level task decision making and reactive motion planning, object recognition and classification in unstructured or uncertain environments. A practical yet effective approach to accomplish these tasks is through human-robot collaboration, in which the robot and human form a partnership and complement each other in performing the task.

We recognize that the human operator excels at determining task goals and recognizing task constraints, if he/she is given sufficient detailed information feedback about the task and the interaction between the tool (e.g., end-effector of the robot) and the task environment. Furthermore, robots excel at being very precise and unaffected by fatigue.

The goal of the research reported in this dissertation is to provide the operator with sufficient information about the task, through visual and haptic feedback, to let the operator define the task model, represented by virtual fixture graphical primitives, through an interactive, or immersive augmented reality interface, and then have the robot actively assist the human operator via virtual fixtures based on the task model.

We evaluate these approaches in both cooperative, or hands-on control and telerobotic control in two different application domains: imaged guided robotic neurosurgery and telerobotic manipulation for satellite servicing tasks under significant time delay.

1.1 Motivation

In neurosurgery operations, a surgeon must accurately locate the relevant anatomical targets such as tumors or bony tissues to be removed, and the critical neurovascular structures, such nerves or blood vessels to be avoided. Further, he/she must perform dexterous manipulation tasks, such as resection of tumors or bony tissues with safety margins of just a few millimeters. Such procedures can often take several hours, causing both mental and physical fatigue that can increase the chance of surgical errors. While recent advances in the use of commercially available navigation systems enabled the intra-operative visualization of registered pre-operative information, e.g., Computed Tomography (CT) or Magnetic Resonance Imaging (MRI)

scans relative to the patient, they cannot actively prevent the surgeon from accidentally damaging critical anatomical structures, such as nerves, blood vessels, or organs during the surgical procedure.

In teleoperation tasks, to ensure successful completion, the operator needs accurate perception of the remote environment, through sensor feedbacks such as visual imagery from remote cameras, or force feedback due to the physical interaction of the remote robot with its environment. These give the operator a sense of "telepresence". However, in the case of ground-based teleoperation of robots for satellite servicing, significant communication delay exists and greatly reduces the operator's sense of telepresence and makes video feedback out-of-sync with reality, and makes force feedback impractical and potentially unstable.

Despite obvious differences in their respective application domains, the neurosurgery and satellite servicing tasks share two common characteristics:

- The human operator must perform precise manipulation with little margin for error.
- The human operator needs accurate and relevant sensor feedback from the task environment to perform the task.

An overview of our approach in each of the tasks is outlined here:

In the neurosurgical application, the surgeon defines a safety boundary, as a virtual fixture task model, on a pre-operative three-dimensional (3D) CT scan of the patient.

After registering this information to the intra-operative coordinate system of the robot, we can use the virtual fixture task model to actively assist the surgeon in a cooperative control setup to prevent surgical errors and reduce operation time.

Uncertainty can prevent the human-robot system from performing the desired task correctly or introduce unacceptable risk. With our collaborators [24], we examined optical sensor based monitoring of patient motion, for compensation of registration displacement. This method is inadequate, however, for measuring tissue deformation, such as tumors internal to an organ, so we propose an intra-operative sensing approach using photo-acoustic ultrasound as future work.

In satellite servicing tasks with time delay, the operator would similarly define a virtual fixture task model, based on desired task goals and geometric abstractions of remote environment. This task model simplifies the interaction between the operator and the remote environment. We can use this task model to provide better visualization, and provide active assistance to the operator through model-based force feedback. We examine methods for using force sensor feedback for remote slave execution, that are robust against modeling and registration errors.

1.2 Dissertation Overview

In Chapter 2, we introduce a common system architecture that uses a virtual fixtures task model for human-robot collaboration for both cooperative and teleop-

eration systems. The prior works for human-robot collaborative systems and virtual fixtures are presented. We then discuss the components of the architecture for both the cooperative control and teleoperation setup.

In Chapter 3, we present the cooperative control implementation of the system architecture for an image-guided robotic system for skull-base drilling. In our approach, the surgeon creates a virtual fixture model by defining the safe zone on a pre-operative Computed Tomography (CT) scan. The virtual fixture is used to constrain the motion of the robot-held cutting tool. We report the results from phantom and cadaver experiments. Last, we discuss the characterization of the optical tracking system measurement noise and patient motion compensation techniques.

In Chapter 4, we present the telerobotic application of the system architecture for satellite servicing tasks, where the human operator performs remote servicing tasks in the presence of uncertainty and telemetry delay of several seconds. We report the development of a new delay tolerant control methodology, using virtual fixtures and force control that is robust to modeling and registration errors. We discuss the technical approach and implementation for both the master and slave side of the telerobotic system. Lastly, we validate the control concept in an application in which the remote slave cuts the tape that secures a multi-layer insulation flap of a satellite mockup.

We conclude our studies and findings in this dissertation in Chapter 5 and address future directions of this work.

1.3 Dissertation Contributions

The major contributions of this dissertation are the following:

1. A system architecture that supports bidirectional information exchange between a human and robot through an interactive graphical specification of collaborative tasks that can be represented by virtual fixture motion constraints.

2. A method to achieve robust execution in the presence of time delay by translating the graphical model to virtual fixtures on the master and a task frame definition for hybrid position/force control on the slave.

3. An implementation of the proposed architecture in the first neurosurgical application of a cooperatively controlled robot with virtual fixtures.

4. An implementation and demonstration of the proposed architecture for a telerobotic system for satellite servicing with time delay.

These are described in subsequent subsections in detail.

1.3.1 A Novel Model Based Robotic Assistance System Architecture

We developed a novel model-based robotic assistance system architecture for both cooperative control and telerobotic systems. It includes an information-enhanced

interactive augmented reality interface that allows the human operator to define and modify virtual fixture task models used by the robot from an environmental model (e.g., from the anatomy) or from a registered geometric model (e.g., line virtual fixture on registered model of a satellite mockup) and provides visual and haptic feedback and mechanical assistance via the robot to the human operator based on the virtual fixture model.

We explored methods for incorporating sensor feedback in the above co-robotic system in order to reduce registration uncertainties through task model updates. For the skullbase drilling application, we examined optical tracking based patient motion compensation methods. In addition, we quantified expected sensor feedback at the slave during the actual task, such as expected cutting force, based on measured static and sliding friction of the cutting surface, for error detection and compensation purposes.

We implemented the above system architecture in two testbeds: in a cooperatively-controlled image-guided robotic system for skull base surgery, and a telerobotic system with an augmented reality interface for satellite servicing tasks under significant time delay.

1.3.2 Skull-base Surgery Application

We developed a cooperative controlled image-guided robotic system for a skull-base drilling task. We employed virtual fixtures generated from anatomy to constrain

the motion of the robot-held cutting tool inside a safe zone defined on preoperative imaging, thus preventing the surgeon from accidentally damaging critical neurovascular structures during the drilling/cutting procedure. We performed experiments on both foam skull and cadaver heads by measuring overcut (i.e., cutting outside the virtual fixture boundary). Through our collaboration with [24], we also explored an optical tracking based patient motion compensation method to improve the accuracy and safety of the system.

1.3.3 Telerobotics for Satellite Servicing Tasks

We developed a telerobotic system for on-orbit satellite servicing tasks in the presence of significant time delay on the order of 4 - 7 seconds. The telerobotic system consists of a modified da Vinci ®Surgical robot master console and a slave manipulator, using a Whole Arm Manipulator (WAM) robot. The user creates a virtual fixture model for the task through an augmented reality interface. During execution, the virtual fixture model provides immediate visual feedback and motion guidance to the operator at the master console, while the remote slave executes commanded motion consistent with these constraints using a local hybrid force/position control scheme. The slave controller also reduces uncertainty due to time delay through model updates, using local intra-operative sensors, such as a 6 DOF force/torque sensor. We also developed reusable components for system infrastructure.

1.4 Related Publications

Sections of the dissertation have been published in conference proceedings and in journals listed below:

Materials in Chapter 3 draw largely on publications in the following:

- T. Xia, C. Baird, G. Jallo, and K. Hayes. An integrated system for planning, navigation and robotic assistance for skull base surgery. *Int. Journal of Medical Robotics and Computed Assisted Surgery*, pages 321–330, Sep 20081

- T. Haidegger, T. Xia, and P. Kazanzides. Accuracy improvement of a neurosurgical robot system. In *Proc. of 2nd IEEE RAS EMBS Int. Conf. on Biomedical Robotics and Biomechatronics*, pages 836 –841, Oct 2008

Materials in Chapter 4 draw largely on publications in the following:

- T. Xia, S. Leonard, I. Kandaswamy, A. Blank, L. L. Whitcomb, and P. Kazanzides. Model-based telerobotic control with virtual fixtures for satellite servicing tasks. In *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, May 2013

- T. Xia, S. Leonard, A. Deguet, L. L. Whitcomb, and P. Kazanzides. Augmented reality environment with virtual fixtures for robotic telemanipulation in space. In *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 5059–5064, Oct 2012

- T. Xia, A. Kapoor, P. Kazanzides, and R. Taylor. A constrained optimization

approach to virtual fixtures for multi-robot collaborative teleoperation. In *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 639 –644, Sep 2011

Chapter 2

System Architecture

2.1 Overview

Human-robot collaborative systems, also known as human-machine collaborative systems (HMCS), are systems that leverage the precision of robotics and the decision making abilities of the human operator to enhance human task performance and avoid the difficulties of fully autonomous robotic execution.

This chapter introduces an unifying system architecture for human-robot collaboration for both cooperative and teleoperative systems, which is applied to the robotic neurosurgery application (Chapter 3) and satellite servicing application (Chapter 4), respectively.

In this chapter, virtual fixture concepts and related work in both cooperative controlled and telerobotic human-robot collaborative systems are presented, followed

by the description and comparison of our proposed system architecture to the related work.

2.2 Background

2.2.1 Virtual Fixtures

Virtual fixtures are computer-generated motion constraints that can provide physical guidance and support by restricting the user's motion in a preferred direction or on a plane, or guiding motion toward a specific location and are often implemented at the core of HMCSs to enhance human performance [3, 59]. They are also known as "active contraints" [15], and "synthetic fixtures" [63]. Virtual fixtures are often categorized as Forbidden Region Virtual Fixture (FRVF), which limits the robot end-effector's motion to a given workspace, or Guidance Virtual Fixture (GVF), which keeps its motion along desired paths or surfaces. Rosenberg [59] first implemented virtual fixtures using impedance planes on the master robot to assist the operator in peg-in-hole tasks on the remote slave.

A number of research studies have shown the efficacy of virtual fixtures in surgical robotics [10, 40, 45] and manufacturing [47]. Funda *et al.* [19] first proposed an optimal control method to control robot systems in a constrained workspace. Li [45] and Kapoor [36] implemented a library of primitives based on geometric task constraints, such as *stay above a plane*, *move along a line*, and *rotate about a point*, that

can be combined using different operators to provide assistance for complex surgical tasks. This approach allows the implementation of both FRVF and GVF and the combination of both for generating complex spatial motion constraints. Bettini *et al.* [10] developed a computer vision based method for generating GVF that provides force feedback to guide a user along a path for retinal applications. Li *et al.* [45] used CT scans of the patient anatomy to generate constraint-based GVF that allow an operator to move a tool along a complex path inside the sinus cavity, while avoiding contact of the tool shaft with the surrounding anatomy. Kragic *et al.* [40] focused on real-time segmentation of user action, recognizing user intention, and then providing appropriate levels of context-specific virtual fixtures. Abbott [3] analyzed the stability of virtual fixtures and introduced the stable pseudo-admittance controller for implementing guidance virtual fixtures on bilateral telemanipulation systems. Lin [47] investigated the portability of virtual fixtures on medical task and manufacturing task on a manufacturing robot.

Compared to these related work, our work in Chapter 3 and Chapter 4 focused on providing a 3D interface that allows the human operator to interactively define and modify graphical virtual fixture geometry primitives which in turn provide visual and haptic feedback and mechanical assistance via the robot to the human operator based on the virtual fixtures. Our virtual fixture implementation in Chapter 4 makes use of the constraint-based approach with virtual fixture parameters extracted from graphical VF primitives.

2.2.2 System Architectures

This section presents the related work on system architectures for cooperative control and telerobotics.

Cooperative Control Systems

In a cooperative robot system, the operator and the robot share the tool and workspace in performing a task. ROBODOC [39, 68], the Johns Hopkins University (JHU) Steady-Hand surgical robot [67], the ACROBOT system [29] and the MAKO Surgical RIO® surgical robotic system [58] are examples of cooperative surgical robot systems.

ROBODOC [39, 68] represented the first use of cooperative control in a surgical application (e.g., in hip replacement surgery), where the surgeon can guide the robot by applying force to the robot's end-effector. Taylor *et al.* [67] proposed the "Surgical Assistant Workstation" (SAW), which emphasized an interactive cooperation between information-driven machines and human surgeons (Fig. 2.1). They developed the Johns Hopkins University (JHU) "Steady Hand" robot, which required human judgement and hand-eye coordinated motion as input, and extended the human's ability to perform micro-scale manipulation tasks with assistance to the operator, which include reducing hand tremor, providing virtual fixture position control and force scaling.

Li *et al.* [45] automatically generated the constraints for robot motion from the CT scans of the patient, and demonstrated the use of the "Steady Hand" robot for

moving instruments inside the sinus cavity. Later iterations of this system have been developed for ophthalmic applications [8, 28]. Kragic *et al* [40] focused on real-time segmentation of user action, recognizing user intention, and then providing appropriate levels of context-specific virtual fixtures. Our work in Chapter 3 is distinct from this system architecture by our support for the interactive virtual fixture definition process, whereas the virtual fixture constraints in these systems are either predefined or generated automatically from anatomy data, or from parsing user motion and understanding user intentions.

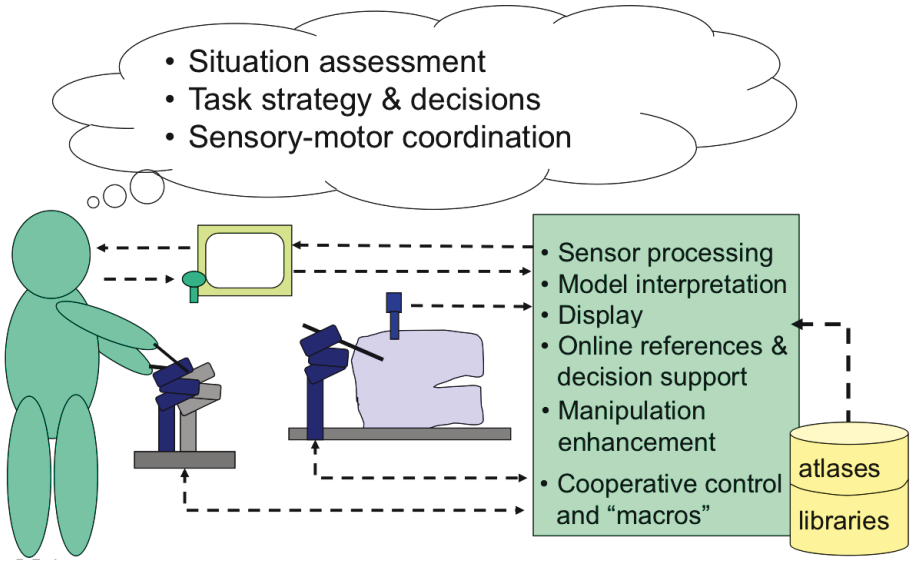


Figure 2.1: Human-machine collaborative systems (HMCS) concept: an interactive cooperation between information-driven machines and human operators that leverages the precision of robotics and the decision making abilities of the human operator to enhance human task performance and avoid the difficulties of fully autonomous robotic execution. This concept supports cooperative control, or teleoperation, or even hand-held control setups. Credit: Russell H. Taylor.

The ACROBOT system [29] is a cooperatively-controlled robot for bone cutting. ACROBOT has been used for Total Knee Arthroplasty (TKA). It employs Active

Constraint Control™ to keep the cutting tool within the region of bone that must be removed to accommodate a knee prosthesis. The knee prosthesis is represented by a 2.5D volume (i.e., a 2D outline extruded in the third dimension).

The MAKO Surgical's RIO® surgical robotic system is used in minimally invasive partial knee replacement procedures, where the surgeon defines a plan for the surgery by placing virtual knee prostheses in a virtual environment of the anatomy created based on the patient's CT scan (Fig. 2.2). When the surgeon uses the robotic arm to manipulate the cutting tools inside the knee, the Tactile Guidance System provides haptic feedback when the surgeon gets close to the boundary of the planned knee replacement [58]. Our work in Chapter 3 is distinct from the ACROBOT system and the RIO system by the different application domain, neurosurgery, vs. orthopaedic surgery.

Telerobotic Systems

In a typical teleoperation system, a human operator manipulates the master-side robotic arm to remotely operate a slave-side robotic arm; the position or force commands are transmitted via a communication channel. Teleoperation systems have been used for handling radioactive materials [21], for lifting and handling large payloads in space using the Canadarm on the International Space Station [5], and for minimally invasive surgery using the da Vinci system from Intuitive Surgical Inc. [23].

In situations with very little or no communication delay, a bilateral teleoperation scheme feedback the environment interaction force at the slave side to the operator to

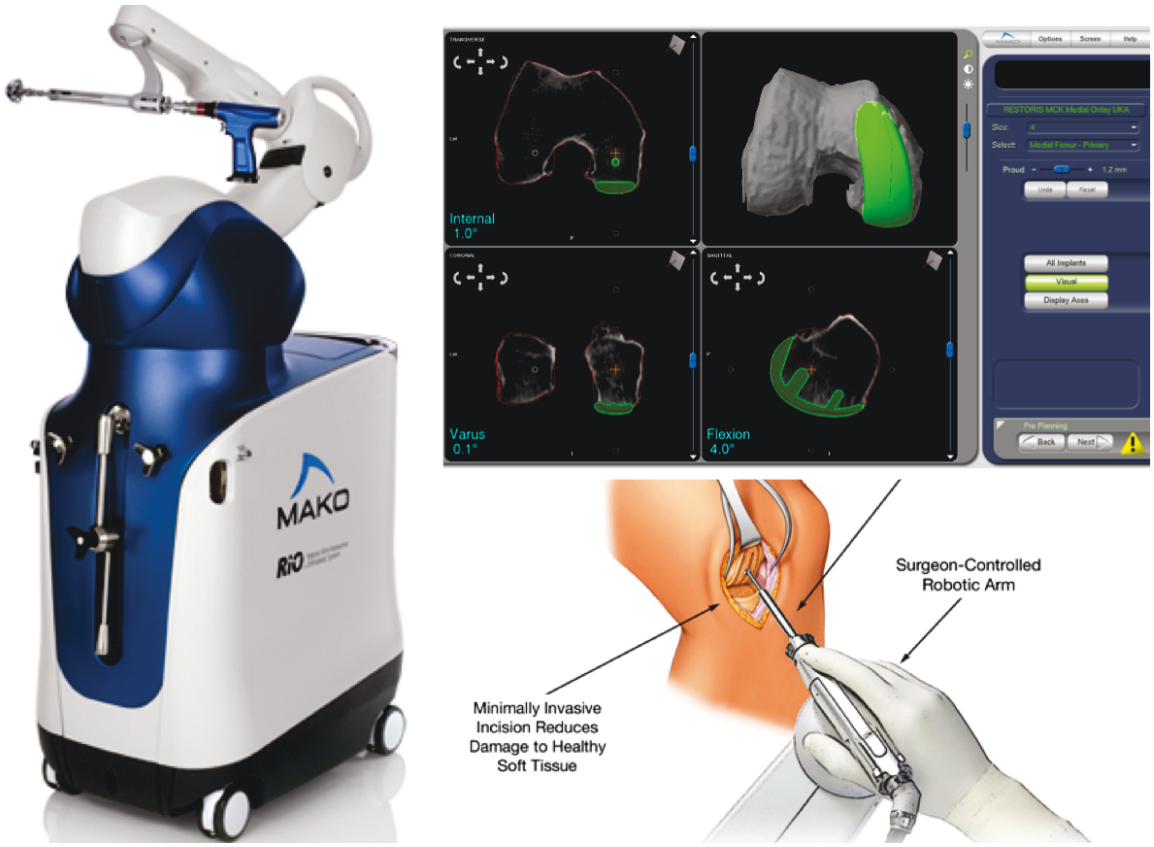


Figure 2.2: MAKO Surgical Corporation's RIO® System.

provide the human operator with better perception of the remote environment [65]. In teleoperation with small or medium delays (fractions of a second), basic force feedback control becomes unstable [26]. Control methods based on maintaining passivity under delay [6] or wave variable transformations of force and motion and power dissipation [55] can achieve stability but are useful for delays under one second [43].

For the satellite servicing application in Chapter 4, transmission delay increases dramatically from ground to orbit due to limits on the speed of light and multiple signal relays at satellite relay stations. According to [65], round trip delays (RTD) between the earth's surface to satellites in low earth orbits are minimally 0.4 seconds, and up to 3 seconds to the moon's surface. With increased delay, action and reaction are out of sync and the operator typically adopts a "move-and-wait" strategy where the operator repeatedly makes small motions, while waiting the round-trip delay for visual feedback from the remote environment to determine the effects of each motion, thus significantly increasing task completion time. Studies performed in the 1960s [17] confirmed that teleoperation task performance is a direct function of the time delay, the ratio of the motion's length and expected task accuracy.

A spectrum of telerobotic techniques have been developed to cope with communication delay. These are related to the basic notion of HMCS because they aim to provide different levels of assistance to the operator depending on context. They generally employ a greater level of abstraction of the transmitted commands and increased autonomy of the remote slave (Fig. 2.3), and range from providing predictive

display [9, 14], shared control [22], to supervisory control [64].

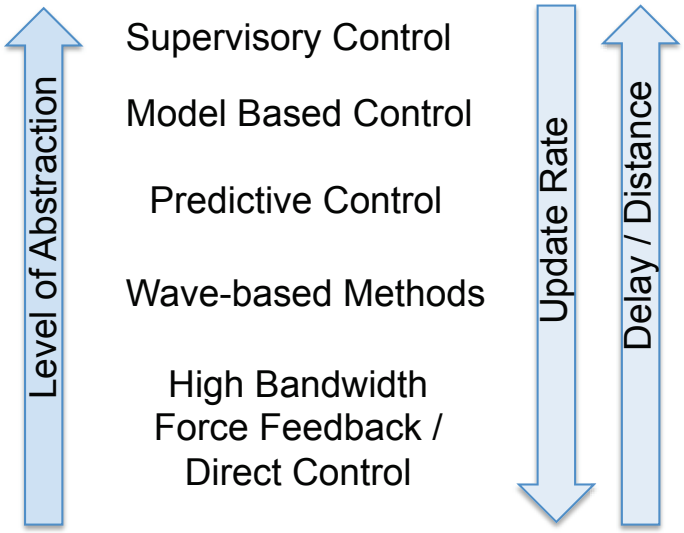


Figure 2.3: Teleoperation techniques with varying levels of abstraction and communication delay. Adapted from [53].

Predictive Display

Predictive display is used to augment video feedback, such that a computer generated representation of the extrapolated position of the remote robot is overlaid on the delayed video feedback [9, 14]. This works well for many unconstrained tasks because it is relatively easy use kinematic and dynamic models to predict where the robot will be a few seconds in the future. However, when the robot must interact with physical objects, it becomes extremely difficult to accurately simulate and predict the robot motion and reaction forces.

Supervisory Control

One proposed approach is supervisory control, where the operator issues high-level goal specifications and intervenes to correct errors, while the remote robot implements

these high-level commands by closing a local high-bandwidth control loop and reports back the task completion status (success or error) at sparse time intervals. Sheridan provided an extensive exposition of supervisory control in [64]. This approach is an option for sufficiently predictable tasks, but suffers from difficulties in automating low-level physical interaction between the robot and the environment at the remote site.

Teleprogramming was first developed by Funda *et al.* [18], and later extended by [27,63]. This approach extends previous supervisory control techniques and shields the operator from the communication delays by automatically generating a sequence of elementary motion primitives based on the operator's action in a simulated environment. After a transmission delay, the slave receives and executes these symbolic motion primitives with real-time local sensor data. Several free space motions and several contact, sliding and pivoting motions can be generated from parsing the operator's interaction with the simulated environment. However, the complexity of motion commands is limited by the difficulty of understanding operator intent and of accurately modeling the environment. Hirzinger *et al.* [27], applied the teleprogramming approach toward the Robot Technology Experiment (ROTEX) on-orbit servicing demonstration mission. Sayers *et al.* [63] applied a similar approach in a subsea teleoperation experiment using acoustic communication. They were able to successfully perform retrieval operations under dynamically changing bandwidth, communication delay and limited computational resources (Fig. 2.4).

While the teleprogramming approach attempts to interpret the operator's action, our approach allows the operator to interactively and explicitly define virtual fixtures and reduce the ambiguity in deciphering the operator's intent. At the same time, virtual fixtures can provide haptic feedback and guide the operator toward deliberate actions (Section 4.3 in Chapter 4).

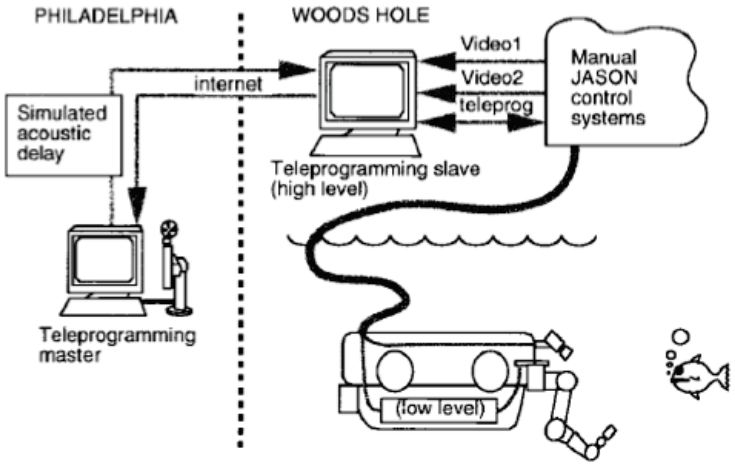


Figure 2.4: Subsea teleoperation experiment with time delay [63].

Model based teleoperation

Approaches where the operator primarily interacts with a model (or simulation) of the task are known as model-based teleoperation. While the teleprogramming approach assumes that a model already exists on the master and sends elemental motion primitives to the slave that are consistent with this model, model-mediated teleoperation [54] extracts the task model from the remote slave's sensor feedback and then sends this model back to the master, where the model can be used to provide

simulated haptic interactions (Fig. 2.5).

Unlike model-mediated teleoperation, our primary goal is to use the sensor feedback to adjust the motion of the slave to better match the model on the master, rather than to update the model on the master. Fundamentally, we assume that most of the time, our model will have the the correct form, but possibly be in the incorrect physical location due to registration errors. It is necessary, however, for the slave to detect situations in which the model itself is inaccurate. We thus develop task monitoring methods that enable the slave controller to quantify the expected sensor feedback, such as the expected cutting force, so that it can detect and potentially compensate for errors without having to wait several seconds for intervention by the human operator.

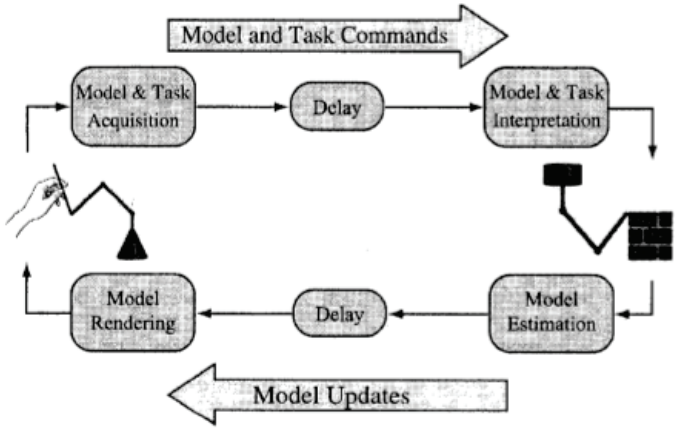


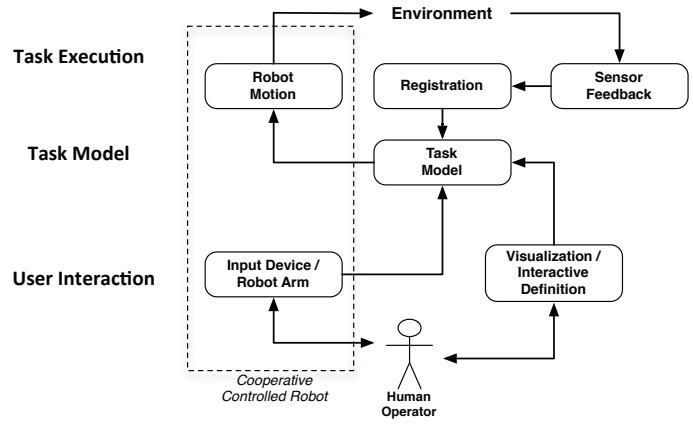
Figure 2.5: Model-mediated teleoperation block diagram [54].

2.3 System Architecture Overview

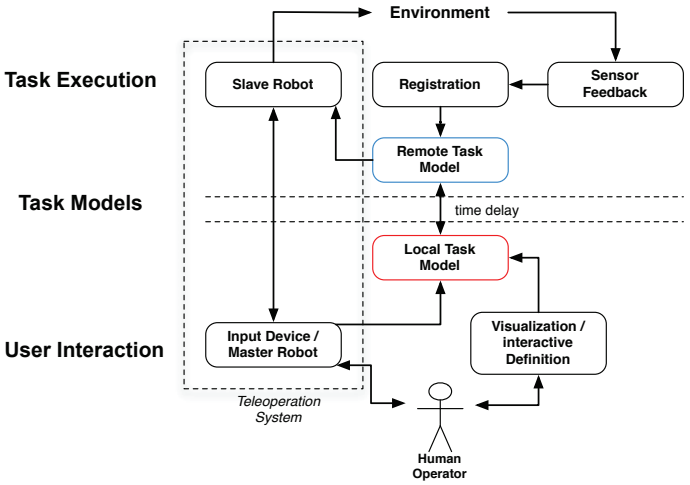
We hypothesize that by providing the operator with adequate information about the task, through visual and haptic feedback, the operator can: (1) define the task model, in terms of task specific virtual fixture constraints and objectives through an interactive, or immersive, augmented reality interface, and (2) have the robot actively assist the operator to enhance the execution time, quality and precision of the tasks.

Fig. 2.6(a) and 2.6(b) show the high-level structure of our human-robot collaborative system architecture, which includes three inter-related problems: *virtual fixtures task model, user interaction to create virtual fixtures*, and *task execution.*

Through an augmented reality user interaction interface, the human operator can intuitively visualize sensor feedback and define or update the task model parameters and robot behaviors; during execution, the human performs the task, but is aided through physical guidance based on virtual fixture motion constraints. Sensor feedback, e.g., end-effector force data, optical sensing, or intra-operative imaging, are used to update and reduce the task model registration uncertainty and to ensure successful task execution.We discuss the major components of the architecture for both the cooperative control and teleoperation systems in the next sections.



(a) Cooperative Control



(b) Teleoperation

Figure 2.6: System architecture components for cooperative control and teleoperation.

2.3.1 Virtual Fixture Task Model

We model the robot as a purely kinematic Cartesian device with tool position $\mathbf{x}_t \in \mathbf{R}^3$ and tool orientation given by unit vector $\hat{\mathbf{l}}_t \in \mathbf{R}^3$, expressed in the robot base frame. The robot has several assigned task frames; for example, an end-effector task frame is assigned at the end-effector and a base frame is assigned at the base of the robot. (Fig. 2.7).

We divide a manipulation task into several subtasks. Kumar [41] and later Kapoor [34] represented each subtask as a distinct state, and used user or sensor input to transition between different states, or subtasks. We are concerned with defining the task model for each subtask, not on the state transitions between subtasks. Each subtask can be modeled using a set of motion primitives based on task geometry, such as "move along a line" and "rotate about a line". For each subtask, we use a task model consisting of several virtual fixtures to specify the desired interaction between the robot and the workspace, where each virtual fixture corresponds to a motion primitive based on the task geometry described above.

A virtual fixture meets the goals of the geometry of the task by modifying the motion of the robot by restricting the free motion along some directions of task frame assigned to a part the robot. For example, a "move along a line" virtual fixture assigned to the robot end-effector will force the end-effector to move along a line. These virtual fixtures are modeled after geometric physical fixture analogues. For each virtual fixture, we create the corresponding graphical primitive, so that the

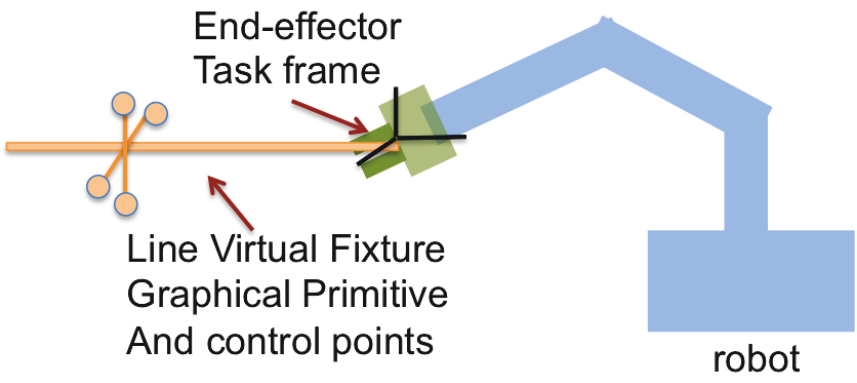


Figure 2.7: Line Virtual Fixture with graphical primitive, control points and assigned task frame on robot end-effector.

operator can visualize and interact with a virtual fixture inside the augmented reality environment (described in the next section). For example, a boundary plane virtual fixture can be defined by drawing a plane in the augmented reality environment. A line virtual fixture is defined by drawing a line, or segments of lines. The graphical primitive has control points such that their associated geometric parameters can be modified by the operator through the graphical interface. For example, a line virtual fixture has Cartesian position and line direction parameters and a plane virtual fixture has Cartesian position and plane normal.

In our skullbase drilling application (Chapter 3), we developed a forbidden region type of virtual fixture by segmenting a pre-operative CT scan, and implemented a control algorithm that slows down the robot-held drill near the boundary and prevents it from moving outside the "safe zone" [70]. A five sided "box-like" convex hull virtual fixture geometry is created interactively by the surgeon and encloses the surface model, then the geometric parameters of the five plane virtual fixtures are

extracted.

In our satellite servicing project (Chapter 4), we have implemented this library of geometric task constraints and describe the details for extracting virtual fixture geometric parameters from their graphical representation in Section 4.3.3. For a teleoperation system with time delay, virtual fixtures can be applied on the master (where there is no delay) or they can be uploaded to the slave robot's controller, in which case they are not affected by the time delay [60, 63].

2.3.2 User Interaction

We recognize that when given sufficient detailed information feedback about the task through visual and haptic feedback, the operator excels at determining task goals and recognizing task constraints and can use this knowledge to interactively define and compose virtual fixtures. We provide this information feedback through an augmented reality user interface consisting of 3D visualization and a 3D user input device that allow the operator to view the task environment and interactively create and/or modify virtual fixtures graphical primitives, thus specifying the task to be performed.

3D Visualization

In the cooperative control setup, the 3D visualization is relatively straightforward. For example, for the neurosurgery application, the 3D visualization is an integrated navigation system that displays the position and orientation of the tracked instrument

(on the robot), relative to a preoperative image (e.g., a CT scan), as well as the visualization of the virtual fixture graphical primitive (Fig. 3.8).

For the telerobotics setup, the visualization maybe also be relatively straightforward; for example, it could consist of a mono or stereo view of the remote environment. But there are situations in which the visualization is more complex. For example, the 3D visualization on the master's side can provide a simulation of the remote slave's interaction with the remote environment under the operator's control, subject to the motion constraints defined by the virtual fixtures model. In addition, the virtual fixture graphical primitive and delayed video feedback can also be displayed (Fig. 4.10).

3D Input Device

The input device is used to interact with the human-machine interface for defining the virtual fixture task model and/or to receive haptic feedback. There are a variety of 3D interaction devices with different degrees-of-freedom and different capabilities, including haptic devices (e.g., Phantom Omni, da Vinci Master-Tool-Manipulator), force sensors and joysticks.

In cooperative control mode, the cooperatively controlled robot arm can be used to interact with the virtual environment or as a separate 6 DOF input device in a "master-as-mouse" mode. If the robot arm is in contact or in proximity to the operation environment, then an external input device might be used to avoid possible collisions, such as a Phantom Omni or a computer mouse.

In the telerobotics control mode, the master arm, when used as n 6 DOF input device, is decoupled from the remote slave arm, and is operated in the "master-as-mouse" mode to interact with the virtual environment. The operator can compose and modify virtual fixture graphical primitives, in a similar fashion to the cooperative control mode.

2.3.3 Task Execution

Once a virtual fixture is defined, it imposes motion constraints on the robot arm and either restricts its motion into forbidden regions, or influences it along a path. In the cooperative control setup, the operator is in contact with the robot arm, and can therefore feel the effects of the motion constraint. In the time-delayed teleoperation setup, the operator receives approximate real-time kinesthetic sensation of the contact interactions between the simulated slave and the surrounding environment subjected to the motion constraint, through the master arm. Thus, the operator is shielded from the destabilizing effects of delayed force feedback. The remote slave executes the time delayed commanded motion consistent with the virtual fixture constraints by using a local, high-bandwidth hybrid force/position control scheme (Section 4.3). This approach is robust against registration errors that exist between the task model and the robot coordinate frame.

2.4 Contributions

We developed a novel model-based robotic assistance system architecture for both cooperative control and telerobotic systems, that supports bidirectional information exchange between a human and robot through an interactive graphical specification of collaborative tasks that can be represented by virtual fixture motion constraints.

Whereas other system architectures generate virtual fixture constraints from anatomical data [29, 45, 58], or from parsing user motion and understanding user intentions [40], the novel aspects of the system architecture include an information enhanced interaction interface consisting of augmented reality, 3D visualization and/or 3D user interaction that allows the human operator to interactively define and modify virtual fixture task models used by the robot from a registered geometric model, and provides visual and haptic feedback and mechanical assistance via the robot to the human operator based on the virtual fixture model. We explored methods for incorporating sensor feedback in the above co-robotic system in order to reduce registration uncertainties through task model updates.

We implemented the above system architecture in two testbeds: a cooperatively-controlled image-guided robotic system for skull base surgery, and a telerobotic system with augmented reality interface for satellite servicing tasks under significant time delay.

Chapter 3

Skull-base Surgery Application

The purpose of this chapter is to present a cooperative control implementation of our proposed system architecture for a skull base surgery task. This chapter is organized as follows. Sections 3.1 and 3.2 describe the motivation and the related work in robotic assistance for neurosurgery. In Section 3.3, we describe the technical approaches of using a virtual fixture model based on the "safe zone" defined on a pre-operative Computed Tomography (CT) scan by the surgeon, to constrain the motion of the robot-held cutting tool. In Section 3.4, we describe the components of the integrated system. In Sections 3.5 and 3.6, we describe the results from experiments on both phantom skull and cadaver heads. In Section 3.7, we discuss the effort to improve the accuracy of the overall system through monitoring for patient motion during the surgical procedure.

This work was done in collaboration with Clint Baird and George Jallo of the De-

partment of Neurosurgery at the Johns Hopkins School of Medicine, Peter Kazanzides at the Laboratory for Computational Sensing and Robotics (LCSR) at the Johns Hopkins University, Kathryn Hayes, Nobuyuki Nakajima, and Nobuhiko Hata of the Surgical Planning Laboratory (SPL) at Brigham and Womens Hospital, and Tamas Haidegger of the Department of Control Engineering and Information Technology at the Budapest University of Technology and Economics. Kathryn Hayes, Nobuyuki Nakajima, and Nobuhiko Hata contributed to the system integration of the intra-operative visualization software, 3D Slicer, with our robotic system (Section 3.4). Clint Baird and George Jallo performed cadaver experiments and collected clinical data (Section 3.6). Tamas Haidegger contributed to methods for improving surgical system accuracy (Section 3.7).

3.1 Introduction

Neurosurgery has undergone tremendous technological innovation over the past half century. The introduction of the operating microscope, stereotactic surgery, modern neuroimaging, neuroendoscopy, technologically demanding implants, and image-guided surgery have enabled advancements while also challenging the limits of human dexterity. The continued advancement of image-guided surgery and the limitations of human dexterity motivate the development of robotic-assisted neurosurgery.

Neurosurgery is well suited to the use of robotic-assisted, image-guided appli-

cations, because the human skull and spine are fixed in place during the surgical operation, and that the human skull has a complex neural and vascular structures. Complications include vascular injury, direct injury to critical neural structures or indirect injury to critical neural structures, through retraction or inadequately gentle surgical manipulation. Sawaya *et al.* reported that neurological deficit occurs in approximately 20% of craniotomies for intraparenchymal brain tumors [62]. Many of these tumors seated in the deep anterior, middle and posterior cranial fossae require complex bone removal for complete tumor resection. Such resection may be complicated by vascular or neural injury. For example, when drilling the posterior wall of the internal auditory canal (IAC) in acoustic neuroma surgery, critical structures such as the semicircular canals, the cochlea, facial nerve and jugular bulb can come within millimeters of the surgical drill. Even when using an established surgical approach, the surgeon may damage the inner ear, vestibular apparatus, adjacent nerves or jugular bulb [48].

While available image guidance systems (Figure 3.1) can use the patient's Computed Tomography (CT) or Magnetic Resonance Imaging (MRI) image scans to provide tool location information and intra-operative anatomical visualization, they cannot overcome the limits of fatigue and dexterity, and cannot prevent or correct the surgeon from accidentally damaging critical anatomical structures.

We developed a system combining intra-operative navigation with robotics to assist with skull base surgery [51,70]. Using a preoperative image, the surgeon delineates

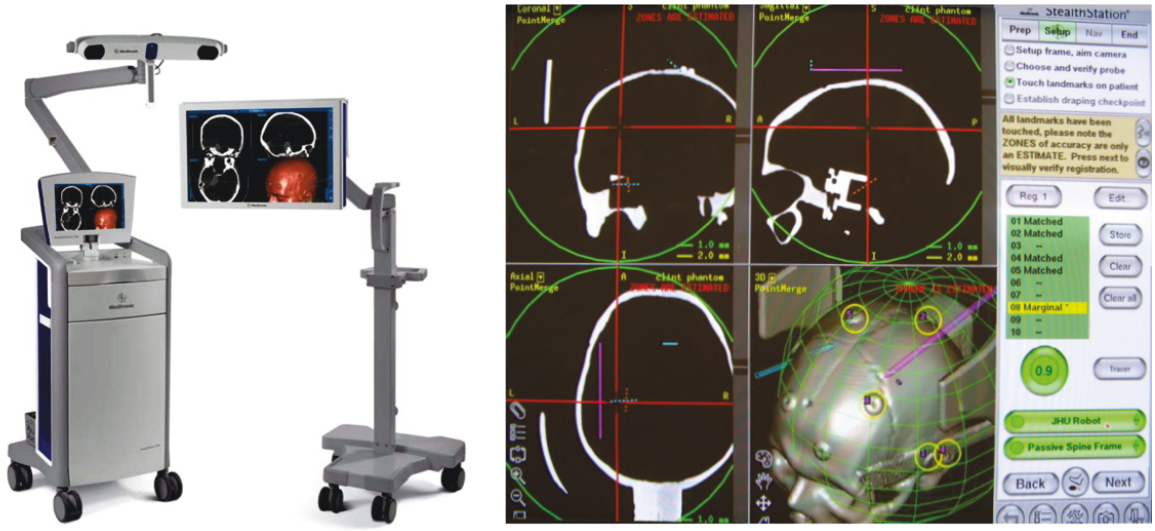


Figure 3.1: (a) StealthStation® Navigation System. (b) Screen capture of the intra-operative navigation user interface.

the portion of the skull base that can be safely drilled, thereby defining a task specification for virtual fixture control that is enforced by the robot. The robot and surgeon share control of the cutting tool in a cooperative control mode, allows the surgeon to retain his/her tactile feedback and sensing of the bone, while the precision of the robot allows the bones to be drilled accurately. This collaborative mode can reduce procedure times and improve patient safety by allowing the surgeon to perform the drilling with confidence that critical structures are protected. A successful drilling procedure is an essential step that enables the surgeon to perform subsequent surgical procedures, such as the resection of tumor tissue, or the placement of stents.

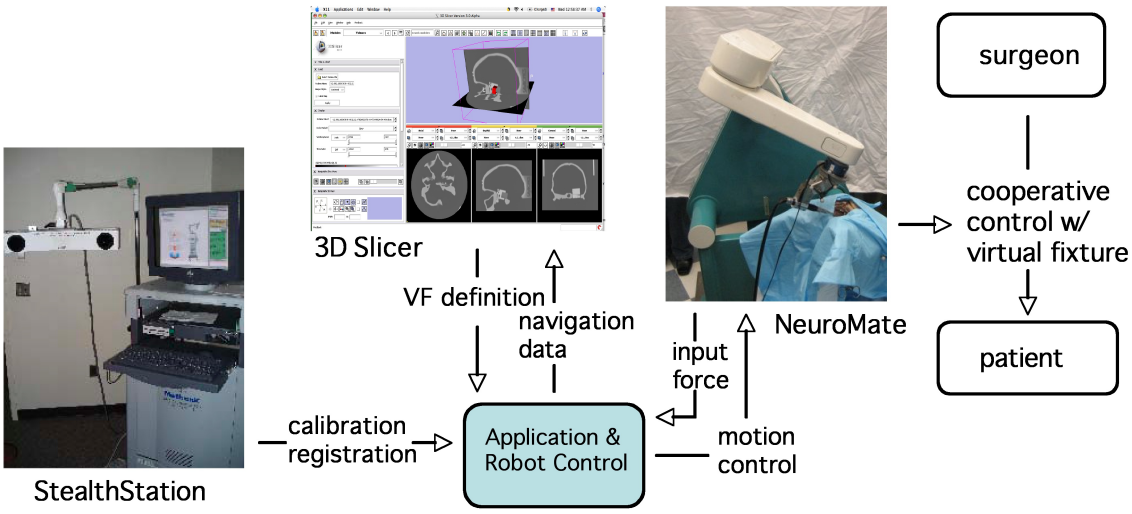


Figure 3.2: System overview of the image guided robot for skull base surgery. System components include (a) modified NeuroMate^{TM} robot in cooperative control mode, (b) StealthStation® Navigation System, (c) 3D Slicer software for intra-operative visualization and (d) PC running the application logic and robot control.

3.2 Related Work

We first review the prior works in robotic systems for neurosurgery: for position and targeting [30, 42, 46] and for drilling the skull [12, 13, 16, 25, 66].

Neurosurgical systems

Robot-assisted neurosurgery started with the develop of robotic systems for holding and targeting biopsy needles. The development of these systems paralleled surgical navigation technologies. After preoperative planning, where the surgeon identifies the target in a preoperative CT/MRI scan, and the robot is registered to the patient's skull, the robot can move an end-effector mounted guide (e.g., needle, probe or catheter insertion) toward the target. Then, the surgeon can manually insert the

guide, or the robot can insert the guide automatically. Kwoh *et al.* [42] used a modified PUMA industrial robot for positioning a biopsy needle. The purpose built 6 DOF NeuroMate robot [46] was used to position brain cannulae, and was the first commercial neurosurgical robotic system to get CE mark in Europe, and US Food and Drug Administration (FDA) approval for the clinical biopsy procedure. Later developments include Minerva [20], which was designed to fit inside a CT scanner, and MARS [30], which is a miniature robot that is directly affixed to a head clamp or to the patient's skull. When compared to using a manually adjustable frame to position the needle guide, the potential advantages of these robotic systems include faster procedure completion time and improved accuracy, assuming proper calibration of the robot.

Several robotic systems were developed for drilling and milling of the skull. Bumm *et al.* [12] developed a modified 6-DoF industrial robot that was integrated with an image-guided navigation system to perform transsphenoidal skull base surgery. The authors reported automated sphenoidotomy and sphenoidectomy operations on cadaveric heads with high accuracy. They reported a mean robot stereotactic accuracy of 1.53 mm. This does not include errors due to the drilling procedure, which were up to 1 mm. The "NeuRobot" [66], used a similar imaged guided approach, where the robot can autonomously remove bone by following the specified path. The innovation of this system includes a preplanning process where the surgeon identifies the "no go" regions on the 2D image slices that are processed to create a Voronoi map that

identifies the largest "go" path. Federspil *et al.* [16] added a force sensor to an industrial robot for bone milling in oto-neurosurgery, in order to prepare the implant bed for a cochlear implant.

3.3 Technical Approaches

The overall technical approach is shown in Figure 3.3. Using a preoperative CT scan of the patient's skull, the surgeon delineates the portion of the skull base that can be safely drilled, thereby defining a task specification for virtual fixture control that is enforced by the robot. The robot and surgeon share control of the cutting tool in a cooperative control mode to perform skull base drilling.

3.3.1 Task Specification

To define the virtual fixture, we use Slicer to segment regions of interest from the CT images and create a surface model (e.g., a Visualization Tool Kit (VTK) polydata file). We simplify the model by creating a six-sided convex hull, removing one or two sides to enable cutter entry. From this geometry object, several planar boundary virtual fixtures are enabled to constrain the motion of the cutting tool (Fig. 3.4). Justification for the simplification is based on clinical input that a "box-like" virtual fixture is sufficient for many skull-base procedures, such as the sub-occipital approach for acoustic neuroma resection, as simulated in our phantom and cadaver experiments.

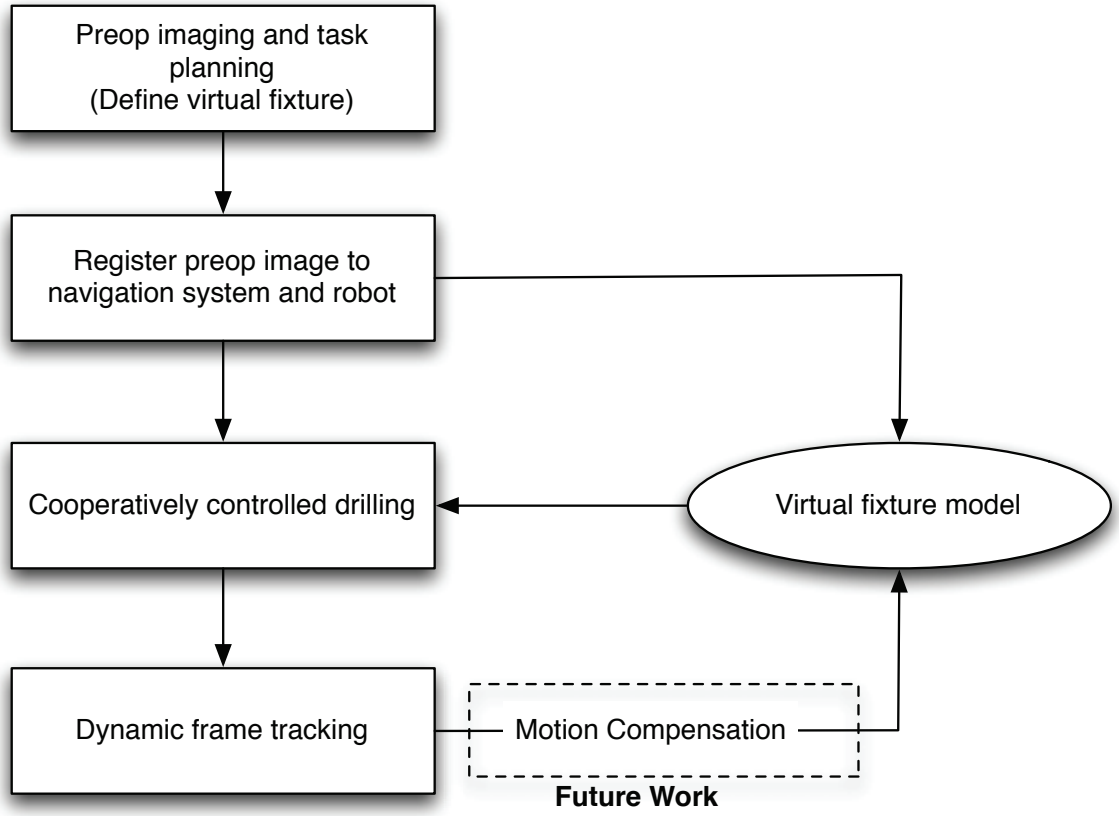


Figure 3.3: Flowchart of skull base drilling procedure workflow.

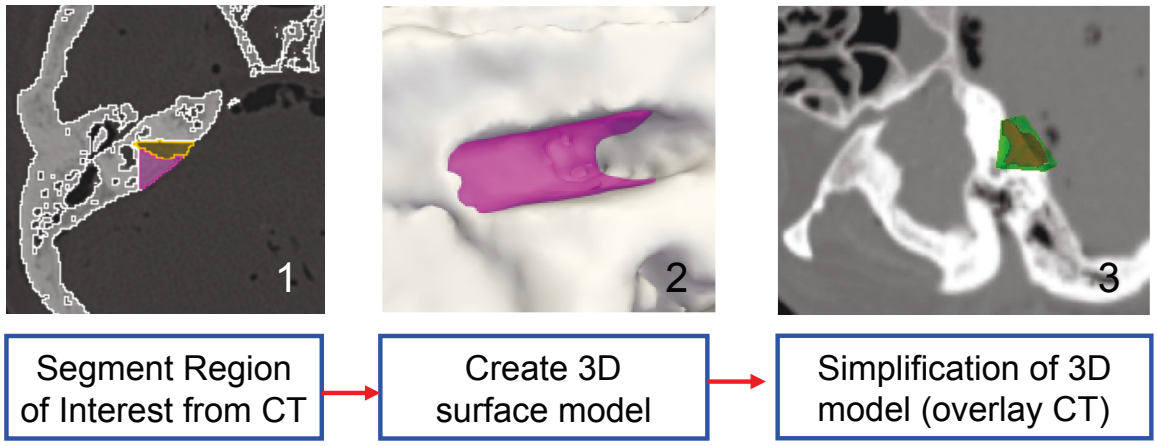


Figure 3.4: Task specification steps: (1) segment safe region from CT slices, (2) create 3D surface model, and (3) simplification of 3D model and overlay on CT.

3.3.2 Virtual Fixture Algorithm Implementation

Virtual fixtures enforce position limitations on the robot manipulator and prevent the robot manipulator from entering into forbidden regions [3]. The workspace of the robot is divided into three regions:

1. A safe zone in which the robot is free to move.

2. A boundary zone between the safe region and the forbidden region. Here, motion of the robot may be restricted, as described below.

3. The forbidden region, which the cutting tool should not penetrate.

The surgeon operates the NeuroMate robot in a cooperative, or "hands-on" control mode, where the readings of the force sensor are converted into motion using an admittance control law (Fig. 3.5). We note that, for convenience, it is possible to

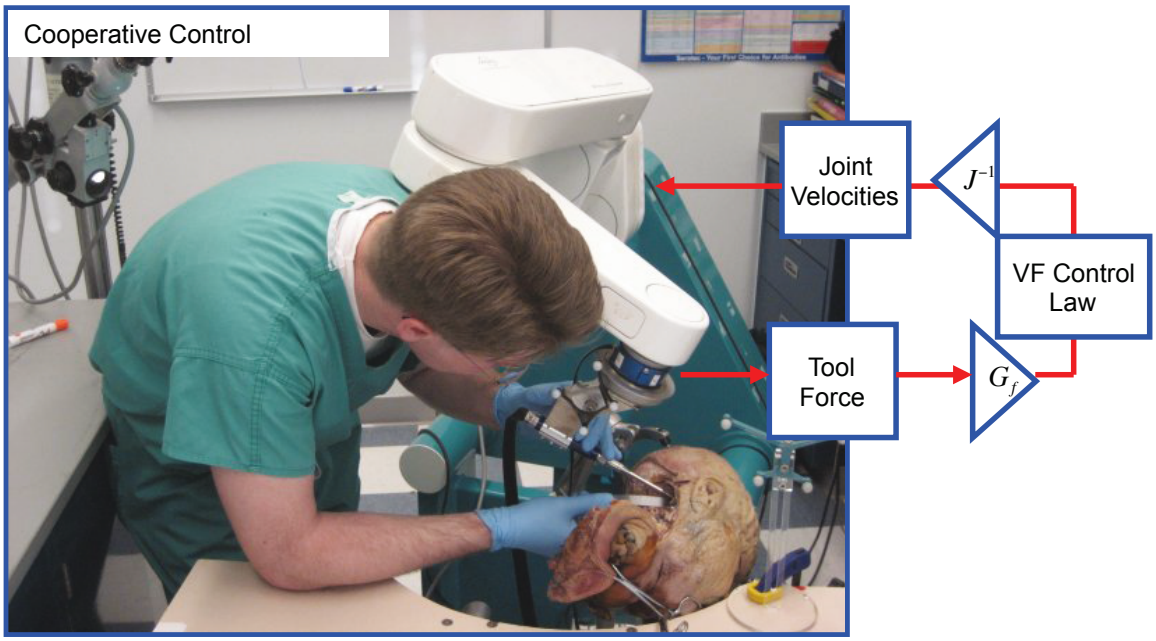


Figure 3.5: Cooperative control of the skullbase drilling robot during a cadaver experiment.

choose translational or rotational motion of the tooltip separately.

The following admittance control law is used:

$$\dot{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{q}) \times \mathbf{K}(\mathbf{d}) \times \mathbf{G}(\mathbf{f}) \times \begin{bmatrix} \mathbf{F_w} \\ \\ \mathbf{T_w} \end{bmatrix} \tag{3.1}$$

where $\dot{\mathbf{q}}$ is the goal velocity in joint space, $\mathbf{K}(\mathbf{d})$ and $\mathbf{G}(\mathbf{f})$ are 6 x 6 diagonal matrices of scale factor and admittance gain, respectively. $\mathbf{J}$ is the Jacobian matrix resolved at the cutter tip. Its inverse, $\mathbf{J}^{-1}$, transforms the Cartesian velocities into joint velocities. $\mathbf{F_w}$ and $\mathbf{T_w}$ form a 6 x 1 vector, and are the measured forces and torques in the Robot World coordinate system.

The admittance gains are nonlinear, and increase exponentially relative to the measured force, $\mathbf{f}$. Large measured forces result in high-speed motion for coarse

positioning of the robot, while lower measured forces result in slow, fine motion control [39]. The parameters for this nonlinear function were experimentally determined. In addition, a deadband near the origin of the measured force $\mathbf{f}$ serves to suppress noisy measurements, while cutoff at high forces keeps the goal velocity below the physical limits of the robotic mechanism.

The virtual fixture algorithm imposes motion constraints by modifying the scale factor $\mathbf{K}(\mathbf{d})$. In the safe zone, $\mathbf{K}(\mathbf{d})$ is set to the identity matrix, so the robot is able to move freely. The boundary zone is defined by a distance, D, from the forbidden zone. The velocity of motions towards the forbidden zone are scaled down by a factor proportional to the computed distance, d, to the forbidden zone. Motions away from the forbidden zone are not modified. If the robot enters the forbidden zone, only motion towards the safe/boundary zone is permitted.

In our current method for computing $\mathbf{K}(\mathbf{d})$, we compute the distance d_i to the virtual fixture planes in each of the *Robot world frame* coordinate directions (i = X, Y, Z). Adopting the convention that the unit normal points in the direction of the "safe zone," the distance d_i is positive if the robot is on the safe side of the plane. If d_i is negative for any plane, the robot is in the forbidden zone, and only motions toward the safe/boundary zone are permitted. If d_i is positive for all planes, the software determines the minimum value, which corresponds to the closest plane in the i direction. If the cutter is moving towards this plane and is within the safety boundary, D, then $\mathbf{K}(\mathbf{d})_i = d_i/D$. This reduces the robot velocity as it approaches

the boundary.

This implementation is effective at preventing the cutter from penetrating the virtual fixture boundary, even when the cutter is at a corner formed by two or more planes. However, it provides a limited ability to move tangential to the virtual fixture boundary.

3.4 Implementation

The system consists of the following major components: a modified NeuroMate robot, a StealthStation Navigation System, a workstation running the 3D Slicer software, and a second workstation running the application logic and high-level robot control (Fig. 3.6).

3.4.1 NeuroMate Robot

The NeuroMate robot, provided by Integrated Surgical Systems in Sacramento, CA, is an FDA cleared image-guided robotic system designed for stereotactic procedures in neurosurgery. The rationale for using this robot includes its mechanical stiffness, good accuracy [46], and convenient workspace for cranial procedures. While the robot was originally designed for positioning and orienting surgical tools, we converted the NeuroMate into a cooperatively-controlled robot by attaching a 6 DoF JR3 force sensor (JR3 Inc., Woodland, CA) at the end-effector, between the final

Figure 3.6: Skull base drilling experiment with cadaver at the Johns Hopkins Medical School's Cancer Research Building. System components include: a modified NeuroMate robot, a StealthStation Navigation System, a workstation running the 3D Slicer software, and a second workstation running the application logic and high-level robot control.

axis and the electrical surgical drilling (Anspach eMax, Palm Beach Gardens, FL). Forces and torques exerted by the surgeon are translated into joint motions to move the instrument in the direction of the applied force. The system can allow unimpeded motion of the instrument or can impose virtual fixtures [4] to guide the surgeon's hand and/or enforce safety constraints, as described in Section 3.3.2. The robot kinematic equations, including tool calibration, provide the location of the cutter tip relative to the Robot World frame (Fig. 3.9).

3.4.2 StealthStation Navigation System

The StealthStation is a commercial navigation system marketed by Medtronic Navigation (Louisville, Colorado). The StealthLink interface allows a third-party client to connect to the StealthStation and acquire data via Ethernet at run-time. The StealthStation tracks the position and orientation of sets of optical markers arranged in a precisely-known geometry, i.e., a *rigid body*. When a rigid body is attached to an instrument, there is an additional calibration step that is required to determine the position and orientation of the instrument tip relative to the coordinate system defined by the rigid body. We adopt the conventional approach of expressing positions and orientations relative to a reference frame (rigid body) attached near the operative site, which defines the Stealth reference frame. This technique is robust with respect to camera motion because the relationship between the tracked instrument and the fixed reference frame would not change (within the accuracy threshold).

We use the StealthStation, with a standard pointer probe, to register the anatomy to the preoperative CT image (i.e., the transformation between the StealthStation reference frame and the Stealth CT frame, Fig. 3.9). Although we used fiducial-based registration for our experiments, the StealthStation also supports anatomic registration. We also mount a rigid body on the robot cutting tool (Fig. 3.9), which enables us to co-register the robot and StealthStation and provides intraoperative visualization of the cutting tool.

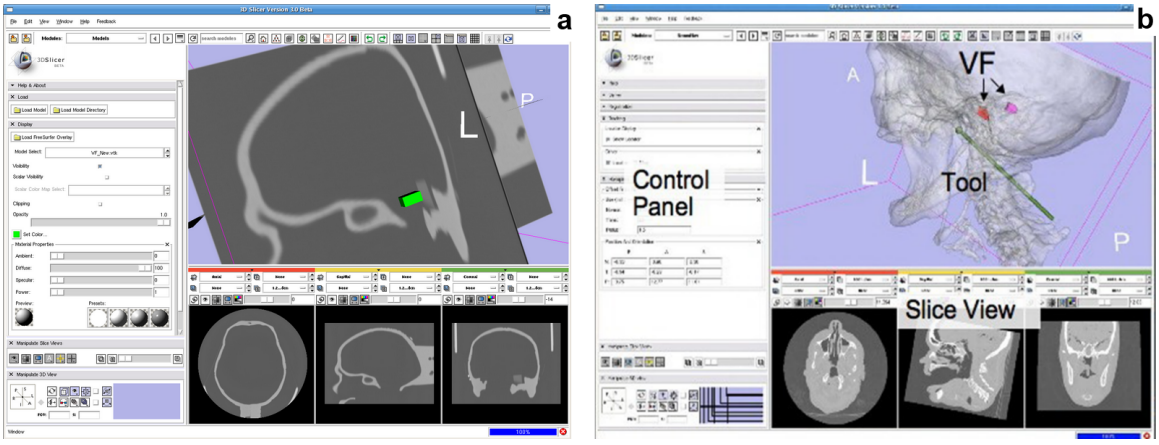


Figure 3.7: Screenshot of the 3D Slicer for intraoperative visualization, loaded with (a) phantom CT, and (b) defined virtual fixtures.

3.4.3 3D Slicer

3D Slicer (www.slicer.org) is an open source, cross-platform application for visualizing and analyzing medical image data (Fig. 3.7). We use Slicer as the planning system because it enables us to create complex virtual fixtures and export them in

an open file format (e.g., VTK polydata). We also use Slicer for intraoperative visualization of the cutting tool with respect to the preoperative CT image because, in contrast to the StealthStation visualization, it displays the 3D model of the virtual fixture and includes a more realistic model of the cutting tool. The robot software provides periodic updates of tool position and orientation to Slicer via a network interface (Fig. 3.8).

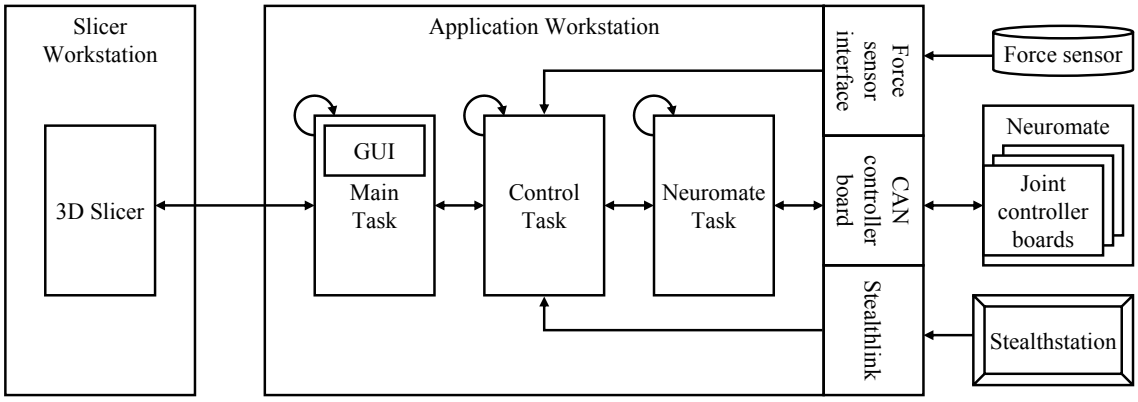


Figure 3.8: System components (left to right): 3D Slicer planning and visualization platform, application logic, robot control and interfaces to force sensor, robot joint controllers, and StealthStation Surgical Navigation System.

3.4.4 Application Controller

The application control software, which includes the high-level robot control, runs on a workstation that contains the Real-Time Application Interface (RTAI, www.rtai.org) for Linux. The software uses the *cisst* package (www.cisst.org/cisst),

an open source medical robot controller framework developed at our research center [35]. The application is partitioned into the *Main*, *Control*, and *Robot* tasks (Fig. 3.8). The *Robot* task communicates with the NeuroMate robot via the Controller Area Network (CAN) bus and performs basic functions such as receiving joint feedback and sending joint setpoints. The *Control* task implements the supervisory control layer. Its primary functions are to provide cooperative force control and virtual fixture computation during drilling. The *Control* task also provides the interfaces to the force sensor and the StealthStation. The *Robot* and *Control* tasks both require periodic, real-time execution, which is provided by RTAI. The *Main* task handles the graphical user interface (GUI), implemented using the Fast Light Toolkit (FLTK, www.fltk.org). The GUI serves as the control panel for the robot and drives the procedure flow. In addition to displaying basic information, it also offers the option to switch between different guidance modes, force or position control, perform calibration/registration or save the motion sequences of the robot. The *Main* task also provides the data to Slicer for intraoperative visualization.

3.4.5 Registration and Calibration

The surgeon uses Slicer to define the "safe zone", also known as the virtual fixture, in the preoperative CT image. This information is loaded into the application control software, which will use it to affect motion in the Robot World frame. Before operation, a series of registration procedures have to be performed. The system

uses several different coordinate systems, as every device has its own frame and this complete set of transformations is shown in Fig. 3.9. Homogeneous coordinate transformations allow us to compute the position and orientation of an arbitrary point in any of the frames once the intermediate transformations are known.

Calibration

Because the virtual fixture is intended to constrain motion of the cutter tip, it is necessary to calibrate the tool so that the cutter tip is known with respect to the Robot world frame and in the StealthStation frame . The robot kinematics already provides the location of the end-effector with respect to the world frame, so it is only necessary to measure the offset (translation) between the origin of the end-effector frame and the cutter tip. Similarly, for the StealthStation, it is only necessary to measure the offset between the cutter tip and the rigid body attached to the robot, through a standard pivot calibration method.

Registration

While the StealthStation and Slicer both read the CT data, they use different conventions for the CT coordinate system; therefore, we require a fixed transformation between the Slicer CT frame and the Stealth CT frame. The transformation between the Stealth CT frame and the StealthStation frame is obtained using registration methods provided by the StealthStation. For the current experiments, we used a fiducial-based registration, where a tracked, hand-held pointer probe is used to touch at least four features (e.g., skin fiducials, cranio-facial screws, or anatomic points)

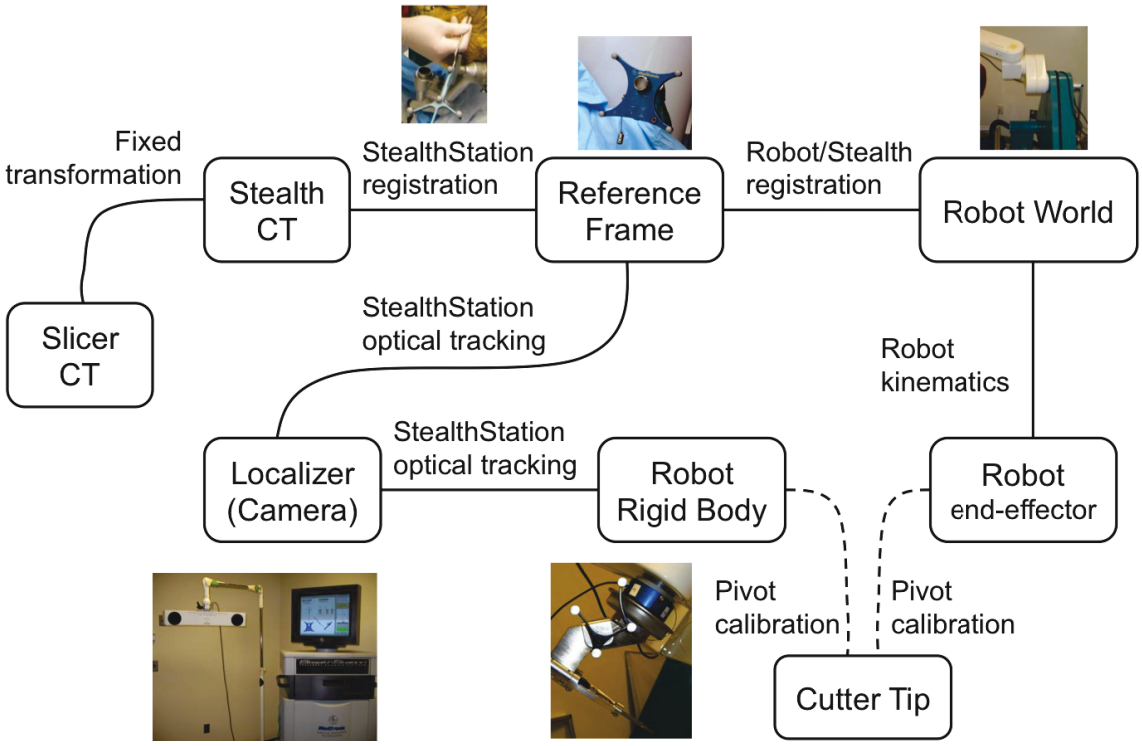


Figure 3.9: Transformation map for the coordinate frames.

attached to the skull prior to the CT scan. The transformation between the Robot World frame and the StealthStation frame is obtained by moving the robot to six different positions, recording the cutter tip position in each coordinate system, and applying a standard paired-point registration method [7, 69].

3.5 Phantom Experiments and Results

3.5.1 Accuracy of Robot and Navigation System

Procedure:

The first set of experiments used a metal plate with small conical divots at different positions and heights (Fig. 3.10). This plate was machined on a Computer Numerical Control (CNC) machine with a known accuracy of 0.0005 inches (0.0127 mm) and was originally used to validate an image-guided robot for small animal research [44]. The test was performed by placing the robot in cooperative control mode and guiding the cutter tip (5 mm diameter sphere) into the neighborhood of each divot. Once the cutter tip was near the divot, the robot controller employed the *ball-in-cone* method [39] to automatically center the cutter tip in the divot. The software then recorded the position of the cutter in the Robot World frame and in the Stealth reference frame.

We characterized the accuracy by computing the Fiducial Registration Error (FRE) [52] and the Fiducial Distance Error (FDE), as described in [38]. The FRE

Figure 3.10: Robot accuracy measurement setup and CNC machined plate with small conical divots.

was computed by registering all thirteen robot (or tracker) positions to the CNC positions, which served as the "gold standard." For the robot, the FRE was 0.64 mm and for the navigation system, the FRE was 0.74 mm. We computed the distances between each pair of points in robot (or tracker) coordinates and compared them to the known distances obtained from the CNC positions. The FDE is the mean difference between the measured and known distances. For the robot, the FDE was 0.49±0.39 mm and for the navigation system, it was 0.53±0.46 mm.

3.5.2 Accuracy of Integrated System

These experiments were performed using a plastic skull with an embedded fixture for inserting foam blocks that represent the target anatomy. The experiments measured the accuracy and repeatability of the integrated system.

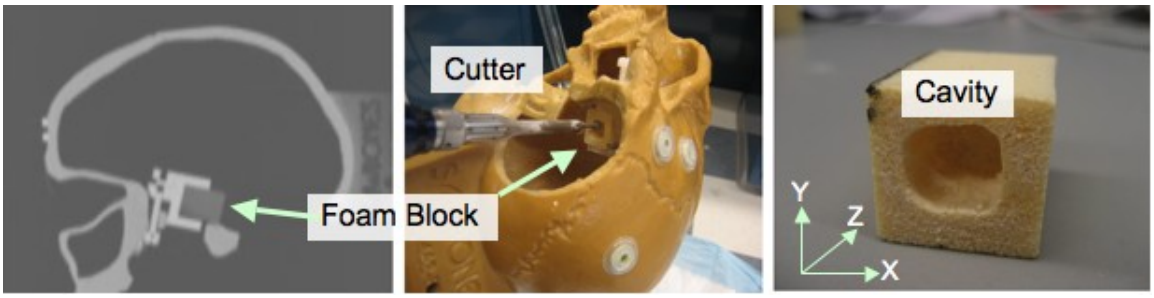


Figure 3.11: Phantom experiments: (left) CT slice view of phantom with foam block, (middle) experiment set up, and (right) machined foam block.

Procedure:

The phantom consists of a plastic skull containing an embedded fixture for holding a precisely machined foam block (Fig. 3.11b). The plastic skull also has several adhesive fiducials attached to its surface. These fiducials are visible in the CT scan and are used during the registration process. The phantom was CT scanned with 2 mm slice spacing (Fig. 3.11a). For the virtual fixture, we defined a box whose edges are offset from the fixture (block) edges by a specific amount. We performed the registrations described Section 3.4.5, and machined the foam block in cooperative-control mode with the robot (Fig. 3.11c). The block was then removed from the fixture and the distances between the machined edges and block edges were measured

using calipers. The experiment was repeated for six foam blocks. For the first three foam blocks, we cut each block using the same registration. For the next three foam blocks, we move the skull into a different position, performed the registration and then cut the foam blocks. In each experiment, we ensured that the cube rested firmly against a marked corner and referenced all measurements to this corner.

Results:

For each machined foam block, we took two measurements of the distance between the machined edges and block edges at different depths of the cut volume and average the measurements. The averaged measurements are summarized in Table 1.

For the X and Y directions, we computed a placement error, E_p, defined by the difference in the centroids of the actual and desired cut volumes. The dimensional error, E_d, is defined by the difference between the actual and desired cut volume dimensions. A positive value of E_d indicates an overcut, i.e., cutting beyond the boundary of the virtual fixture. The total overcut error, due to both placement and dimensional error, is equal to $|E_p| + E_d/2$.

The mean and standard deviation for trials 1, 2 and 3 are "Mean1" and "SD1". The low values of SD1 (less than 0.25 mm) demonstrate that the robot system had excellent repeatability when the same registration was used to machine the first three blocks. For the four foam blocks cut with different registrations, the results are "Mean2" and "SD2". These latter results are more representative of the overall system performance, although additional trials should be performed to achieve statistical

significance.

3.6 Cadaver Experiments and Results

Cadaver experiments were performed at the Johns Hopkins Medical Institute (JHMI) to verify the system with a more complex VF, and to gain insight into the emerging difficulties of a more realistic setup. A neurosurgery resident performed the experiments and provided valuable feedback on the system for further development. An appropriate VF was created for the resection of skull bone to treat a hypothetic acoustic neuorma via suboccipital approach (Fig. 3.12). The typical bone tissue resected was around 0.2 to 1 cm^3.

The resident performed this procedure on the left and right sides of three cadaver heads, for a total of six trials. The first trial was unsuccessful due to an implementation error and was excluded from further analysis. We obtained and analyzed postoperative CT scans for three of the remaining five trials.

3.6.1 Procedure

For each specimen, we generated the virtual fixture for the internal auditory canal (IAC) from a CT scan of the cadaver head at 0.5mm slice spacing, through manual segmentation (a combination of thresholding and freehand drawing) using 3D Slicer. The virtual fixture encompasses the IAC's posterior wall (Fig. 3.13). The virtual

Table 3.1: Results of phantom experiments (errors in mm). The placement error, E_p, is defined by the difference in the centroids of the actual and desired cut volumes. The dimensional error, E_d, is defined by the difference between the actual and desired cut volume dimensions. Dimensional error is positive for overcut (more bone removed).

Foam	E_p		E_d		**Depth**
	X	**Y**	**X**	**Y**	**Z**
1	0.17	1.12	0.54	0.25	1.16
2	0.04	1.08	0.50	0.20	1.06
3	0.49	0.96	0.25	0.05	1.19
Mean1	**0.23**	**1.05**	**0.43**	**0.17**	**1.14**
SD1	**0.23**	**0.09**	**0.16**	**0.10**	**0.07**
3	0.49	0.96	0.25	0.05	1.19
4	1.28	1.11	0.70	0.33	0.51
5	-0.44	0.79	0.99	0.35	1.39
6	1.04	-0.62	0.54	0.10	1.85
Mean2	**0.59**	**0.56**	**0.62**	**0.21**	**1.23**
SD2	**0.76**	**0.80**	**0.31**	**0.15**	**0.56**

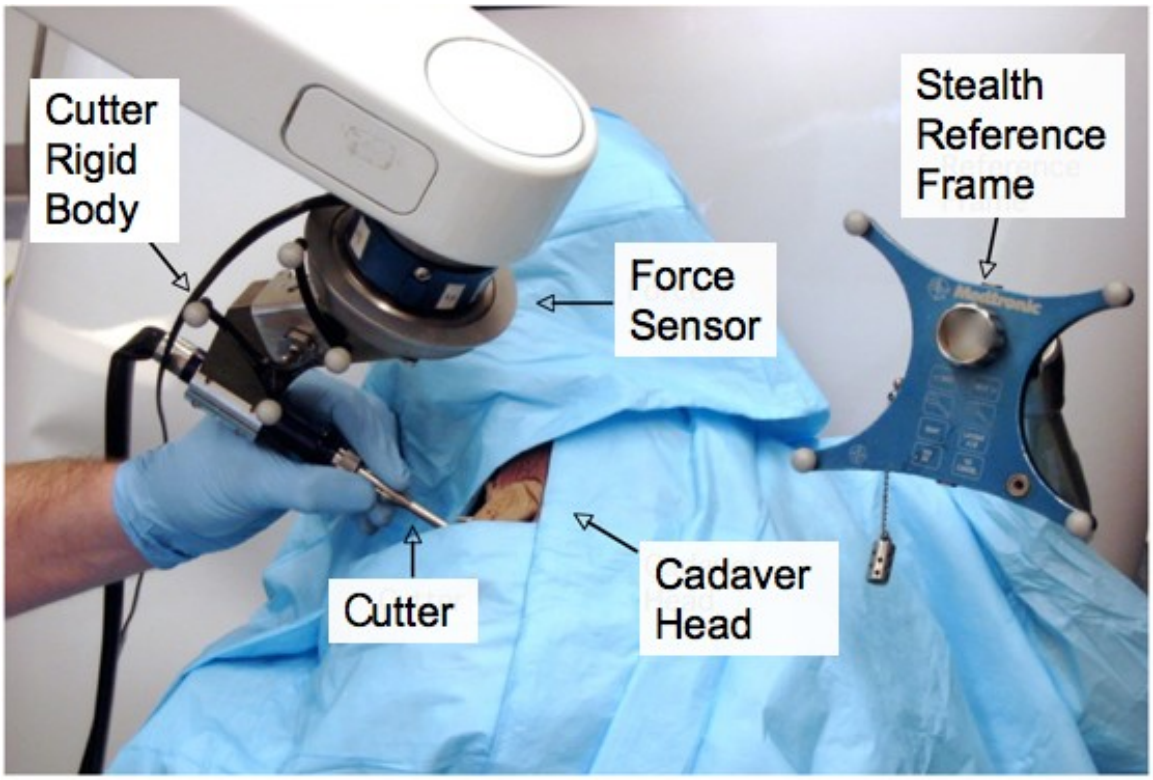


Figure 3.12: Setup for cadaver experiment. The surgeon operates the robot-mounted surgical drill in cooperative control mode.

fixture was shaped so that we could observe the anatomical structure of the fundus as typically done in a real surgical procedure. Prior to positioning into the robot field, a retrosigmoid craniotomy using standard surgical techniques gained access to the cerebellopontine angle. The cadaver specimen was secured to a three point Mayfield skull clamp in the lateral position.

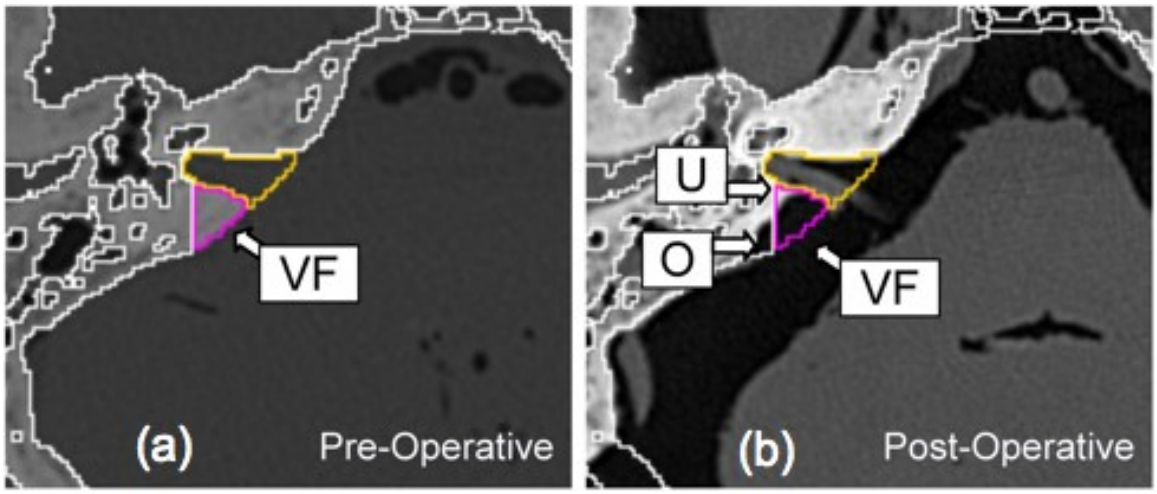


Figure 3.13: **Specimen 1: (a)** pre-operative CT cross-section showing virtual fixture (VF). **(b)** post-operative CT cross-section showing uncut bone (U) and overcut (O).

3.6.2 Results

The NeuroMate proved to have sufficient workspace and dexterity to perform the procedure, which is not surprising given that this robot was designed for neurosurgical procedures. It was, however, necessary to carefully position the robot with respect to the head to avoid the kinematic singularities, where it becomes difficult to control the robot. The neurosurgeon operating the system noted that it improved the efficiency

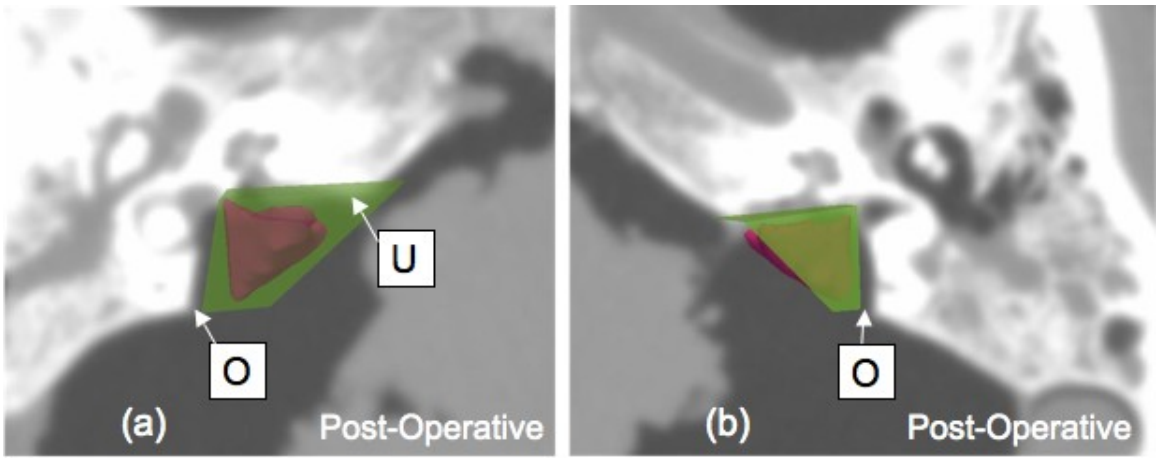


Figure 3.14: **Specimen 2: (a)** post-operative CT with VFs (original and simplified) in place showing the overcut (O) and uncut bone (U) of the left side IAC. **(b)** the overcut (O) of the right side IAC procedure.

and ergonomics by stabilizing the drill and by maintaining its position when released.

We used 3D Slicer to manually align the preoperative and postoperative CT scans for two cadaver heads, where the drilling procedure was performed on one side of the first head and on both sides of the second head. We then transformed the preoperatively-defined virtual fixtures (and simplified convex hulls) to the postoperative CT scans. This enabled us to visualize both uncut bone (i.e., bone inside the virtual fixture that was not cut) and overcut bone (i.e., bone outside the virtual fixture that was cut). Fig. 3.13 and Fig. 3.14 show representative 2D cross-sections and 3D views, respectively, for the two cadaver heads. There were areas of uncut bone (U), but we note that the objective of this procedure is to remove enough bone to access the tumor – it is not clinically necessary to remove all bone inside the virtual fixture. A more critical measure is the amount of overcut bone (O), because this can

Table 3.2: Registration residual errors in cadaver experiments. Residual errors (a) StealthStation to CT residual errors, and (b) StealthStation to Robot residual errors, in millimeters.

Trial No.	Cadaver	Procedure	Residual error (a)	Residual error (b)
1	A	Left Porus	0.61	0.36
2	B	Right Porus	0.86	0.48
3	B	Left Porus	0.94	0.33

affect the safety of the procedure if the overcut area includes critical neurovascular structures. We measured the overcut in several CT cross-sections for both specimens and found that it was typically 1-2 mm, with occasional excursions up to 3 mm. The Mayfield clamp was secured to the base of the NeuroMate robot to avoid relative motion.

We used the StealthStation to register the StealthStation reference frame to the Stealth CT frame. We only accepted the registration if the residual error displayed by the StealthStation was less than 1 mm. We then registered the Stealth reference frame to the Robot world frame, as described above, and only accepted this registration if the residual error (FRE) was less than 0.5 mm (see Table 3.2).

3.6.3 Discussion

The cadaver experiments indicate that a cooperatively-controlled robot system could feasibly be used in a clinical setting, although a significant amount of engi-

neering effort would be required to bring this prototype to clinical use. In these experiments, the cutting tool often penetrated the virtual fixture by 1-2 mm, with occasional excursions up to 3 mm. The phantom experiments produced better results, most likely due to the more favorable experimental conditions and the use of foam rather than bone. These experiments showed a mean placement error of 0.6 mm and a mean dimensional error of 0.6 mm, which implies a mean overcut error of about 0.9 mm. There are many possible causes of placement error, including registration error (CT to Stealthstation and Stealthstation to robot), calibration error (cutter to robot and cutter to StealthStation), robot kinematic error, and undetected motion of the skull. The most likely causes of dimensional error are compliance in the system and anomalies in the machining process (such as the "imperfect drilling characteristics" noted in [12]), but robot kinematic error or skull motion can also be factors. One approach for reducing the effect of system compliance is to use telemanipulation rather than cooperative control because it eliminates the deflection due to surgeon-applied forces, as demonstrated in [45]. On the other hand, telemanipulation requires additional hardware and may not provide the surgeon with the same feeling of direct control.

3.7 Accuracy improvement

Efforts to improve the accuracy of the overall system include improving the accuracy of the robot and compensating for patient motion during the surgical procedure. This study is included to show that the system architecture support intra-operative registration update based on sensor feedback to compensate for patient motion.

3.7.1 Improved Fixation

Mechanical improvements to the mounting platform on the robot base (where the head clamp is attached) produced encouraging results. We repeated the robot and navigation system accuracy testing described above, with the aluminum plate attached to the new mounting platform. In this configuration, the mean fiducial registration errors (FREs) for the robot and navigation system were 0.38 mm and 0.45 mm, respectively, which represent about a 40% improvement over their respective prior values of 0.64 mm and 0.74 mm. We hypothesized that this improved fixation would have produced better results in the cadaver experiments. Furthermore, even with rigid fixation, it is still advisable to at least monitor, and preferably track, relative motion between the patient and robot, as discussed in the next section.

3.7.2 Patient motion monitoring

In the previous experimental setup, the virtual fixture computation was performed in the Robot world frame after registration. This had the advantage that it does not require line-of-sight between the StealthStation localizer and the robot during the procedure. The disadvantage is that this method transforms the virtual fixture to the Robot world frame at the start of the procedure and does not detect or correct deviations from the initial transformation. Deviations occur when the patient skull is moved, or the surgeon leans against the skull during surgery and result in the displacement of the virtual fixture relative to the skull. If the virtual fixture computation is instead performed in the Stealth reference frame using the position of the tool tip that is tracked by the StealthStation localizer, then we can compensate for unintentional patient motion (Fig. 3.9). However, this technique requires a clear line of sight, and is adversely affected by noisy StealthStation localization measurements, as compared to the robot encoder measurements, which contain only quantization noise.

Haidegger *et al.* [24] characterized the StealthStation localizer measurement noise and the robot intrinsic accuracy. The author assisted in robot software development and data collection for these experiments. Localizer measurements were collected while a high-precision 3 axis linear robot (New England Affiliated Technologies, Lawrence, MA) was actuated to move in straight line trajectories. Both Kalman filters and a moving-average filter (2 second window) were tested with the system and

used to smooth the StealthStation measurements [24].

For motion compensation, we mounted the Reference Frame directly on the skull or on the cranial frame (Fig. 3.9). Using the StealthStation, we can monitor the relative position of the two frames (Reference Frame and Robot Rigid Body), and can detect and compensate for unintentional motions of the patient with respect to the robot and update the virtual fixture registration. We validated the feasibility of this approach through a series of experiments [24], but have not repeated the cadaver experiments.

3.8 Chapter Summary

This study reports the development of a cooperatively-controlled robotic system for skull base surgery with virtual fixture motion constraints. The system has the potential to enable surgeons to more quickly perform skull base drilling while improving safety by preventing the surgeon from accidentally damaging critical anatomical structures during the drilling procedure.

The placement and dimensional errors were measured in phantom experiments, where the robot drilled box shapes in foam blocks which were subsequently measured using calipers. The placement error is the difference in the centroids of the desired and measured box shapes; its mean value was 0.6 mm. The mean dimensional error was 0.6 mm, which means that the linear dimensions of the box cut with the robot

were 0.6 mm larger than desired.

In the cadaver experiments, accuracy was measured by registering a postoperative CT to the preoperative CT and identifying the overcut areas, which are areas where bone outside the virtual fixture was (erroneously) drilled. Qualitatively, the mean overcut appeared to be between 1-2 mm, with a maximum value near 3 mm. As a future work, we could also collect data for unaided experiments, where the robot is cooperatively controlled by the surgeon, with image guidance, but without the virtual fixture motion constraints. We can compare the accuracy of these unaided experiments to the previous experiments and observe the effect of enforcing virtual fixture constraints.

Areas for improvement include tools for postoperative assessment that can characterize the placement and dimensional error by comparing 3D models of the virtual fixture (from preoperative CT) and the cut cavity (from postoperative CT).

3.9 Contributions

Our robotic system represents the first application of a cooperatively controlled robot in neurosurgery and demonstrates that the proposed architecture is applicable for a cooperatively controlled robot and for a surgical application.

The virtual fixture task model specification process, and the virtual fixture algorithm implementation developed can be generalized and applied to other image guided

surgical procedures, such as laminectomy. For laminectomy, a possible enhancement is to provide a means for the surgeon to adjust the virtual fixture intraoperatively to advance the drilling depth along a single (1D) axis.

In this application, we did not provide a means for the surgeon to adjust the virtual fixture intra-operatively. In the current implementation, the surgeon must segment the safe region in CT using 3D Slicer, run convex hull simplification on the surface model in a separate program and then load the convex hull into the StealthStation. This motivated us to create an online, interactive process to define virtual fixtures, which was used for the satellite servicing application in Chapter 4.

Chapter 4

Telerobotics for Satellite Servicing Tasks

This chapter reports a telerobotics application of our proposed system architecture for satellite servicing tasks. We are developing methods for telerobotic on-orbit servicing of spacecraft under ground-based supervisory control of human operators to perform tasks in the presence of uncertainty and telemetry time delay of several seconds.

We propose a new delay tolerant control methodology, using virtual fixtures, hybrid position/force control, task frame formalism, and environment modeling, that is robust against modeling and registration errors. The task model is represented by graphical primitives and virtual fixtures on the teleoperation master and realized by a hybrid position/force controller on the slave robot. The virtual fixtures guide

the operator through a model-based simulation of the task, and the goal of the slave controller is to reproduce this simulation (after a few seconds of delay) or, if measurements are not consistent with the models, to stop motion and alert the operator. This approach is suitable for tasks in unstructured environments, such as servicing of existing on-orbit spacecraft that were not designed for servicing.

This chapter is organized as follows: Section 4.1 introduces the satellite servicing tasks, Section 4.2 discusses prior approaches to time delayed teleoperation, Section 4.3 explains the overall control architecture's components, Section 4.4, Section 4.5 and Section 4.6 describe the physical implementation of the control architecture and the experimental validations on the ground-based telerobotic testbed, respectively.

This work was done in collaboration with Simon Leonard, Anton Deguet, Isha Kandaswamy, Jonathan Bohren, Kelleher Guerin, Amy Blank, Zihan Chen, Louis Whitcomb and Peter Kazanzides at the Laboratory for Computational Sensing and Robotics at the Johns Hopkins University, Thomas Evans of the Robotics Center at the West Virginia University, and Brian Roberts of the Satellite Servicing Capabilities Office at the NASA Goddard Space Flight Center. Isha Kandaswamy developed the cutting strategies for the thermal blanket cutting task in Section 4.3.6. Several people contributed to the implementation of the telerobotic testbed in Section 4.4: Simon Leonard and Anton Deguet developed the video capture, video and telemetry delay component, and integration of the WAM robot. Jonathan Bohren and Kelleher Guerin designed several essential mechanical parts for mounting sensors on the teler-

obotic testbed. Additionally, Simon Leonard, Amy Blank and Isha Kandaswamy also contributed to the experiment and data collection in Section 4.5 and Section 4.6.

4.1 Introduction

Successful missions to repair and upgrade spacecraft, such as the Solar Maximum Mission (SMM) and the Hubble Space Telescope Servicing Missions demonstrated the value of on-orbit repair and upgrade of an existing satellite [2]. While these missions have fulfilled their objectives, the risk and cost of sending humans in space to service satellites remains the biggest hurdle. With the retirement of the space shuttle program, the United States has lost its national capability to perform on-orbit repair of satellites in low-earth orbit. Also, the space shuttle was not designed to reach satellites in higher orbits. To address these challenges, NASA and the space industry are aiming to service satellites by using robotic spacecraft that will be, in part, teleoperated from earth.

NASA Goddard Space Flight Center's (GFSC) Satellite Servicing Capabilities Office (SSCO) is currently developing and operating the Robotic Refueling Mission (RRM). The goal of the mission is to demonstrate a telerobotic system on the International Space Station (ISS) that, under operator control from the ground, can successfully perform many of the manipulation tasks necessary for refueling a commercial communications satellite [2]. The RRM module contains several task boards

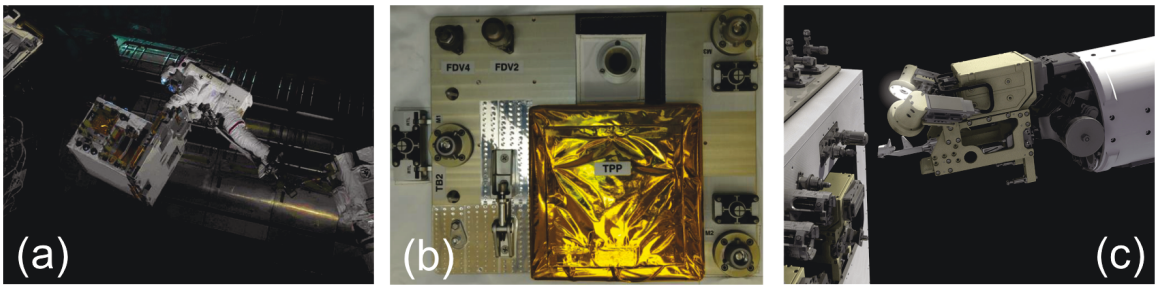


Figure 4.1: (a) The RRM payload module contains several mockup task boards for installation at the International Space Station. (b) A mockup of the MLI blanket covered refuel port. (c) An artist rendering of a RRM tool mounted on the Dextre performing the wire cutting task. Image credit: NASA Goddard Space Flight Center.

and refueling ports resembling a legacy spacecraft not designed for refueling. The Special Purpose Dexterous Manipulator (SPDM), also known as Dextre (onboard the ISS), will use special tool attachments to demonstrate steps in the refueling procedure, including removal of the protective thermal blanket and safety wires, unscrewing and accessing valves, and fuel transfer (Figure 4.1).

In collaboration with GSFC, we report advances in methods for performing a specific satellite servicing intervention task — the cutting of multi-layer insulation (MLI) that normally covers a satellite to protect it from the harsh temperatures in space. Typically, a satellite's fueling port is covered by a rectangular "flap" of MLI that is taped down on four sides (Fig. 4.2). To begin a servicing or refueling operation, the tape that binds the MLI flap must be cut on three sides and then peeled back without damaging the MLI or the satellite. This task is extremely challenging to perform under teleoperation with a time delay of several seconds.

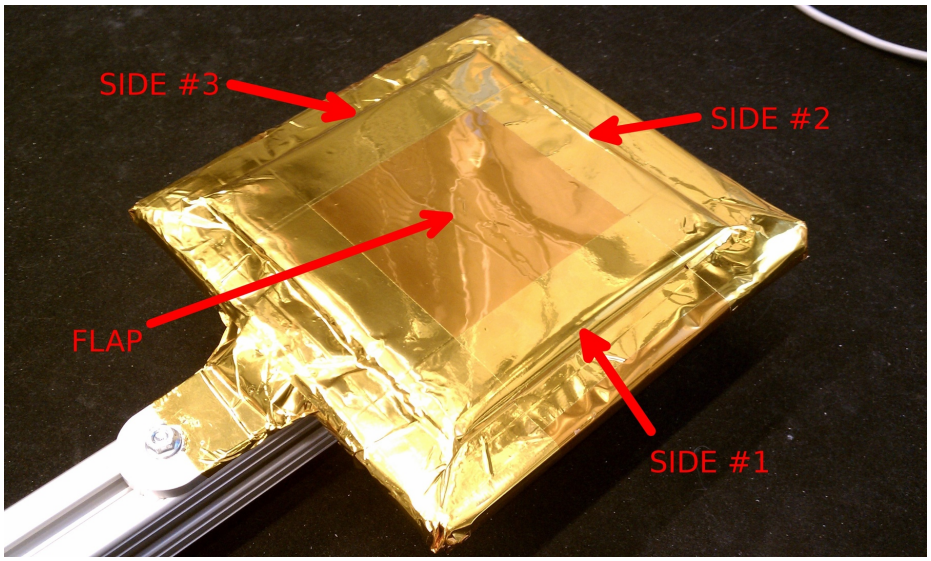


Figure 4.2: Mockup of MLI flap covering a fuel receptacle. The flap is taped down to the surface on all sides with Kapton tape.

4.2 Related Work

Previous on-orbit servicing demonstration missions such as the Engineering Test Satellite VII (ETS-VII) [56] and the Robot Technology Experiment (ROTEX) [27], have demonstrated approaches for telerobotic execution of tasks with significant time delays. We will discuss the control approaches for dealing with time delay in the Section 2.2.2 in the context of these missions.

The Robot Technology Experiment (ROTEX) was an on-orbit servicing demonstration mission developed by the German Aerospace Agency (DLR) and launched into orbit inside the Space Shuttle Columbia (STS-55) in 1993 [27]. Several operation modes were evaluated for the tasks of mechanical assembly, grasping of a floating object and disconnecting/connecting an electrical plug, using the multi-sensory robotic

arm inside a closed work cell. For teleoperation modes with communication delay, ROTEX used predictive display, and tele-sensor-programming where the human operator interacted with a simulated world, and controller sent commands for sensor based execution later on-board. The tele-sensor-programming approach relied on two key features of this system: the multi-sensory gripper and the local sensory feedback loops. Under this approach, operators were able to perform manipulation tasks with up to 7 seconds of delay.

The Engineering Test Satellite VII (ETS-VII) was launched in 1997 by the National Space Development Agency of Japan (NASDA) for the demonstration of robotics technologies in unmanned orbital operation and servicing tasks [56]. The mission consists of two objectives: autonomous rendezvous/dock, where a pair of satellites: a carrier (main) satellite and a target satellite docked together autonomously, and robot servicing experiments with round trip transmission delay of 5-7 seconds. The main satellite is equipped with a task board and a manipulator arm. Ground based operators successfully performed surface following and peg-in-hole tasks. The technical approaches used include predictive computer graphics, which shows a preview of how the manipulator arm will move before sending the commands, and the virtual force feedback to the operator through a haptic device. These approaches were demonstrated to be more effective than conventional move-and-wait teleoperation approaches [74].

4.3 Technical Approach

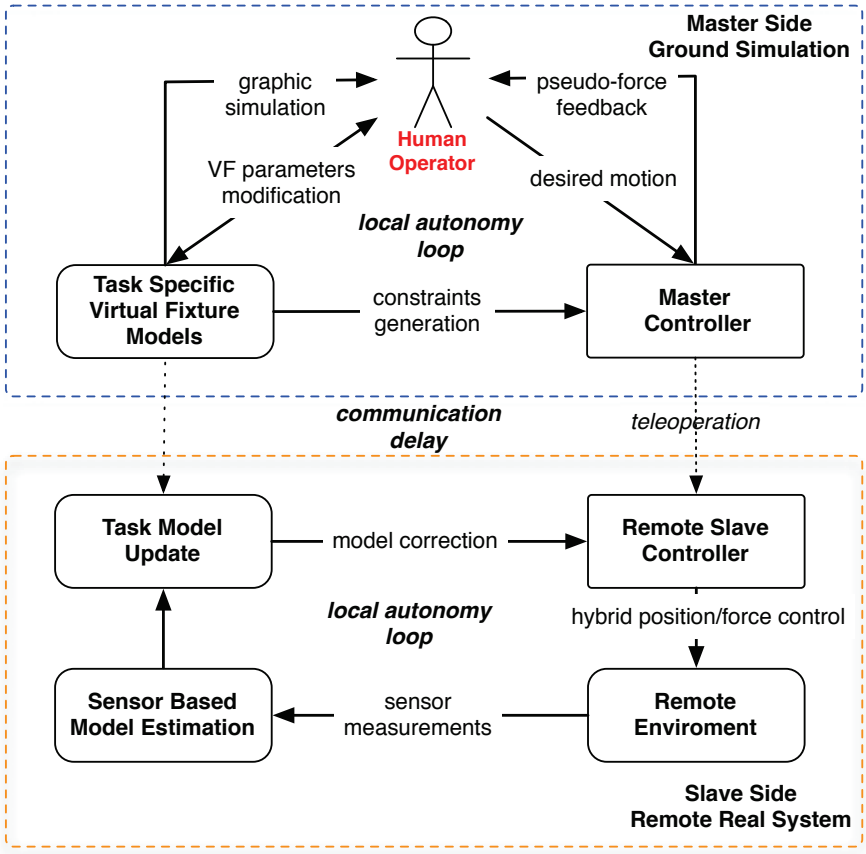


Figure 4.3: Telerobotic System Architecture. The human operator interacts with the simulated environment through an augmented reality interface that provides visual (graphical simulation) and kinesthetic (force) feedback and can reprogram the controller through virtual fixture modifications.

Our technical approach is an extension to the teleprogramming and the model-mediated teleoperation approaches and combines aspects of these prior approaches (Fig. 4.3). We propose to create and update models on both the master and the slave. We assume that for many candidate tasks, there is sufficient *á priori* information (e.g.,

from satellite CAD drawings) to create an initial virtual fixture model, which can be later revised by the operator via an augmented reality interface on the master console and used to guide the motion during the teleoperation.

Our initial work [72] assumed that the remote environment did not change or deform and that the registration between the remote slave and the remote environment was accurately known. In practice, task geometric uncertainties may exist and can lead to large contact forces and moments. We incorporate sensor feedback at the remote site to mitigate problems caused by registration errors. The remote slave executes the time delayed commanded motion consistent with the virtual fixture constraints by using local, high bandwidth hybrid force/position control. Furthermore, we developed task monitoring methods that enable the slave controller to quantify the expected sensor feedback, such as the expected cutting force, so that it can detect and potentially compensate for errors without having to wait several seconds for intervention by the human operator.

The telerobotic system's block diagram is illustrated in Fig. 4.4. The user creates a high-level model *Task Model* by defining and positioning graphical primitives, such as lines and planes, and associated data in a simulated environment on the *Master Display* (which could be augmented with delayed video feedback). Then, this *Task Model* is transformed into virtual fixtures for guiding the motion during the teleoperation, and provides immediate visual and kinesthetic feedback to the operator on the *Master Manipulator*. This *Task Model* is also transformed to task frames for *Hybrid*

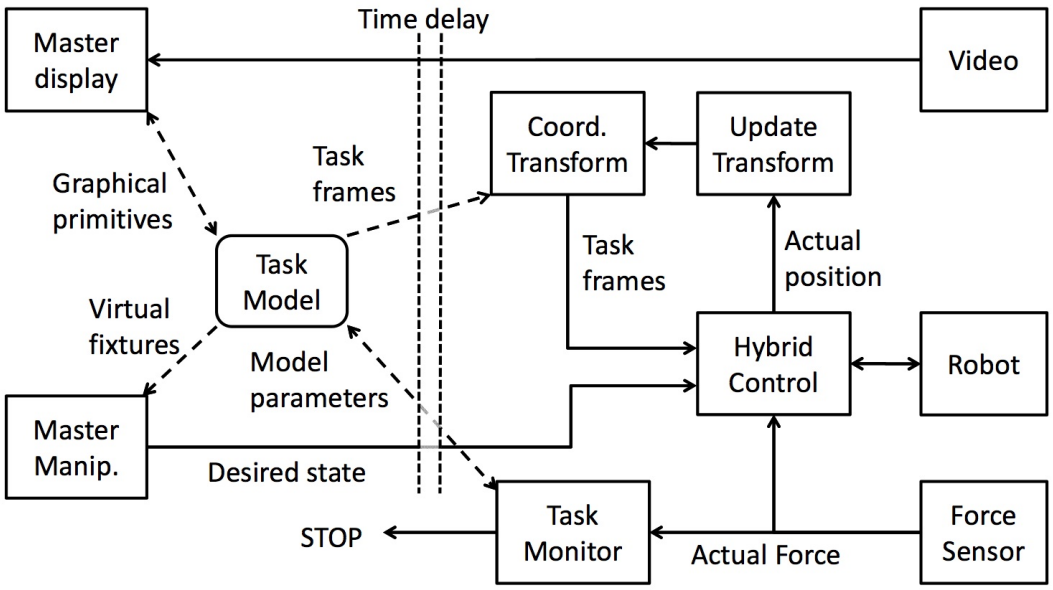


Figure 4.4: Block diagram of the system. The operator creates a *Task Model*, which is represented by virtual fixtures on the *Master Manipulator* and task frames on the slave robot *Hybrid Control*. Measured sensor feedback can be used by a *Task Monitor* to ensure consistency with the task model (or update the task model) or by *Update Transform* to correctly position the task frame.

Control on the slave robot.

The task frames must be expressed in the slave's coordinate system, which requires a coordinate transformation (*Coord. Transform*). The slave executes the time delayed commanded motion using local, high-bandwidth hybrid force/position control. This transformation may not be accurate, but it can be updated based on sensor feedback (*Update Transform*). The task model also provides parameters that are used by a *Task Monitor* to verify that the measured sensor feedback is consistent with the assumed task model and possibly to update the model based on the sensor feedback. Both the *Task Monitor* and the *Update Transform* can use sensor feedback to adjust the *Task Model* parameters or its registration to the physical world, respectively.

The following sections present further details on the *Task Model* (for cutting MLI tape), the virtual fixture control on the *Master Manipulator*, the *Hybrid Control* on the slave robot, and the *Task Monitor* on the slave robot.

4.3.1 Task Model for Cutting MLI Tape

The MLI manipulation task can be decomposed into the "approach", "puncture" and "tape cutting " subtasks. We focus on the last subtask because it is considered to be the most difficult and time consuming to perform without assistance. For this subtask, the operator commands the slave robot to cut along the seam of the tape that secures the MLI blanket on the spacecraft (Figure 4.2) and must avoid bunching up the tape, pulling on the MLI with excessive force and tearing it, or applying excessive force against the spacecraft.

We have developed two cutting strategies [33], (1) the *compression-based strategy* and (2) the *tension-based strategy*, based on the manner in which the cutter interacts with the tape. In the first strategy, the robot exerts a downward force while cutting, which compresses the tape against the MLI on the surface of the satellite. In the second strategy, the robot pulls up, thereby applying tension to the tape while cutting. This second strategy is necessary for cases in which there is no satellite surface beneath the tape. Using the initial model of the satellite's computer-aided design (CAD) drawing and MLI flap, we create a task model represented by a combination of primitive objects, such as planes and lines. From this high-level task model, we

generate the appropriate information for the various modules in our system, as shown in Fig. 4.4 and described in the following sections.

4.3.2 Virtual Fixtures for Master Manipulator

On the master side, we developed an approach that combines elements of augmented reality for visualization with virtual fixtures for guiding the motion during the teleoperation. A virtual fixture performs a function analogous to that of a physical fixture (e.g., a ruler) and can be adjusted in a virtual environment at run time. Using virtual fixtures in the context of a teleoperation task results in a system that combines the accuracy of a robotic system and the expertise of a human operator with improved safety and efficiency [19, 45, 70]. This synergy is especially important for systems with large time delays because the virtual fixture can be applied on the master, where there is no time delay [59]. Alternatively, if the virtual fixture is defined relative to environment features and the slave robot has sensors to detect these features, the virtual fixture can be uploaded to the slave robot's controller and avoid the effect of time delay.

For the tape cutting application, we can use "stay above a plane" or "move along a line" virtual fixture primitives [72]. Fig. 4.5 shows two planar virtual fixtures: the lower one can be used for the compression-based strategy, whereas the upper one can be used for the tension-based strategy. If used together, they constrain the cutter within the small region between the planes. These planar virtual fixtures do

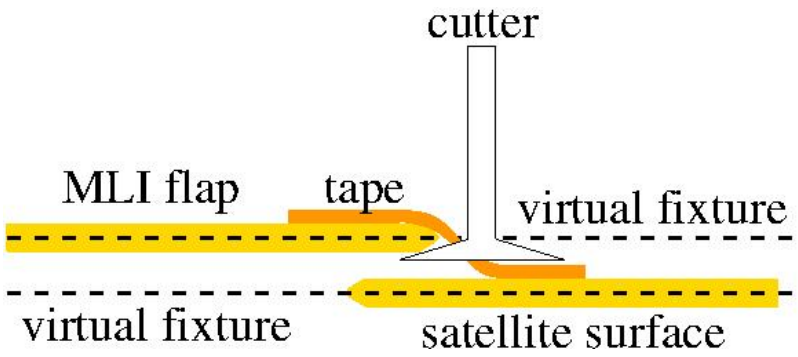


Figure 4.5: Cross section view of the satellite surface. The cutter must make an incision and slide along the flap without cutting the MLI blanket or damaging the satellite. Planar virtual fixtures ensure that the cutter remains within a reasonable cutting height [33].

not constrain the lateral motion of the cutter with respect to the tape seam, so that control is left to the operator. Other task models and virtual fixtures could be defined to further constrain the system.

4.3.3 Virtual Fixture Constraints

Kapoor *et al.* [36] proposed an approach for implementing virtual fixtures, which formalizes virtual fixture primitives as spatial motion constraints in a weighted, multi-objective constraint optimization framework. As the operator performs a small motion in the virtual environment, the virtual fixture controller computes an optimal incremental motion $\Delta\mathbf{x}_m$ based on the operator's desired incremental motion $\Delta\mathbf{x}_m^d$. The optimization objective minimizes the difference between the optimal (also known as the "commanded") master incremental motion, $\Delta\mathbf{x}_m$, and the desired master incremental motion, $\Delta\mathbf{x}_m^d$. The computation of $\Delta\mathbf{x}_m$ is expressed as a quadratic con-

strained optimization problem where a virtual fixture primitive is represented by linear inequality constraint(s) of the form:

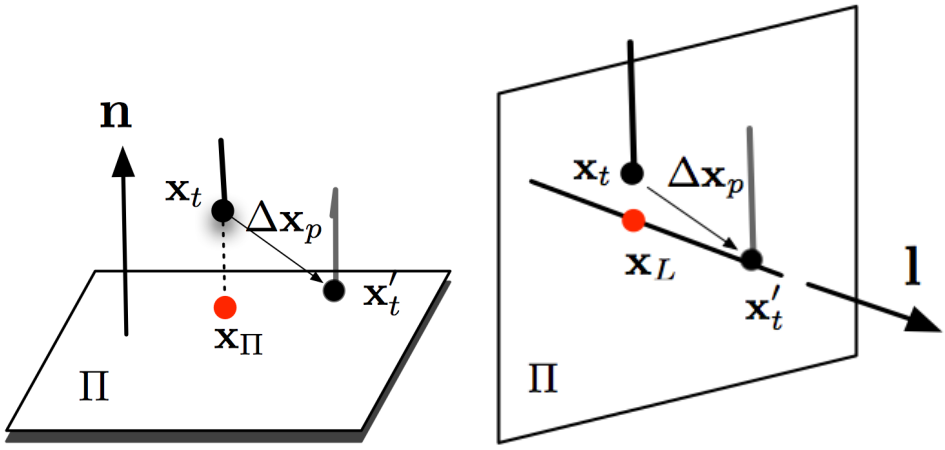
$$\begin{aligned} \min_{\Delta \mathbf{x}_m} \ & \left\| \Delta \mathbf{x}_m - \Delta \mathbf{x}_m^d \right\|^2 \\ & h_1(\Delta \mathbf{x}_m) \leq 0 \\ s.t. \quad & \vdots \\ & h_N(\Delta \mathbf{x}_m) \leq 0 \end{aligned} \tag{4.1}$$

where $h_1, \ldots, h_N$ represent the constraints of N virtual fixtures. For teleoperation, the optimal incremental motion, $\Delta \mathbf{x}_m$, is transformed into the slave coordinate system, $\Delta \mathbf{x}_s$, and executed by the remote slave robot after the transmission time delay.

The primary role of the virtual fixture primitives is to modify the motion of the robot by restricting the free motion of the frame involved. For the planar virtual fixture, the nominal behavior is to keep the remote robot's tool frame position, $\mathbf{x}_t$, from penetrating the given plane, Π, that has the unit normal direction of $\mathbf{n}$ pointing to the free half space and passing through the point $\mathbf{x}_c$ on plane Π, which is closest to $\mathbf{x}_t$ (Fig. 4.6(a)). The tool is free to move in the free half space, proportional to the operator incremental motion, $\Delta \mathbf{x}_m^d$, in position and orientation. For the planar virtual fixture defined by a plane Π, we then define the inequality constraint

$$h_\Pi = \mathbf{n}^T \cdot ((\mathbf{x}_t - \mathbf{x}_c) + \Delta \mathbf{x}_m) - \epsilon \leq 0 \tag{4.2}$$

where $\mathbf{x}_c$ is the closest point on plane Π to $\mathbf{x}_t$. The terms used in both Equation 4.2 and Equation 4.3 are summarized in Table 4.1 and Table 4.2.



(a) Plane constraint (b) Line constraint

Figure 4.6: Virtual fixture geometry models illustration. $\mathbf{x}'_t$ is the desired tool tip position at time step $t + \Delta t$. $\Delta\mathbf{x}_p$ is the operator's desired incremental motion at the master. The terms used in this figure are summarized in Table 4.1 and Table 4.2.

Table 4.1: Summary of Terms Used: Plane Virtual Fixture

Term	**Definition**
$\mathbf{n}$	Plane normal
$\mathbf{x}_c$	Closest point on plane Π to $\mathbf{x}_t$
$\mathbf{x}_t$	Tool frame position
$\Delta\mathbf{x}_m$	Incremental motion
$\mathbf{x}'_t$	Tool frame position after incremental motion

Table 4.2: Summary of Terms Used: Line Virtual Fixture

Term	**Definition**
$\mathbf{l}$	Line direction
$\mathbf{x}_c$	Closest point on the line $\mathbf{L}$ to $\mathbf{x}_t$
$\mathbf{x}_t$	Tool frame position
$\Delta\mathbf{x}_m$	Incremental motion
$\mathbf{x}'_t$	Tool frame position after incremental motion

For the line virtual fixture, the behavior is to keep the tool position $\mathbf{x}_t$ moving along a reference line L, which has the direction $\mathbf{l}$ and passes through point $\mathbf{p}$; the movement should be proportional to the operator input $\Delta\mathbf{x}$, as shown in Fig. 4.6(b). For each primitive, the operator also defines a small positive tolerance value, ϵ, that serves as an absolute bound for deviation from the nominal behavior. For the line virtual fixture defined by a reference line L in 3D space, given by $L : L(s) = \mathbf{p}+\mathbf{l}\cdot s, s \in \mathbb{R}$, we define the inequality constraint

$$h_L = \left\| ((\mathbf{x}_t - \mathbf{x}_c) + \Delta\mathbf{x}_m)) \parallel \Pi_L \right\| - \epsilon \leq 0 \tag{4.3}$$

where $(\mathbf{x}_t - \mathbf{x}_c) + \Delta\mathbf{x}_m \parallel \Pi_L$ represents the projection of the tool frame motion onto the plane Π_L that is perpendicular to line L. This shows the distance between the tool frame after the incremental motion, $\mathbf{x}'_t$ and the closest point, $\mathbf{x}_c$ on L.

4.3.4 Augmented Reality for Virtual Fixture Modeling

In our blanket cutting task, the 3D augmented environment is composed of the delayed visual feedback, the graphical models of the satellite and the virtual fixtures used for the task (Fig. 4.7). The 3D model of a satellite can be obtained by either using data provided by its manufacturer or by using a camera or range sensor from the docking spacecraft [72]. The model is registered in the coordinate system of the remote slave arm and overlaid in the augmented reality environment.

Using the virtual environment, the operator instantiates the necessary virtual fixtures and adjusts their geometric parameters. In this "master-as-mouse" mode, the remote slave arm is "clutched" from the master arm and the operator only interacts with the virtual environment. The operator can drag and rotate the 3D interaction markers (IMs), thereby applying rigid body transformations to the geometry associated with each virtual fixture (see the IMs in Fig. 4.7). The associated geometric parameters, such as plane normal $\mathbf{n}$ and point $\mathbf{p}$ can be extracted. The operator can interactively add or remove the virtual fixtures.

Master Desired Incremental Motion

We need to compute the operator's desired incremental motion $\Delta\mathbf{x}_m^d$ (Equation 4.1) for the optimization objective function on the master manipulator [34]. For an admittance-type manipulator, which generally has high friction and inertia and

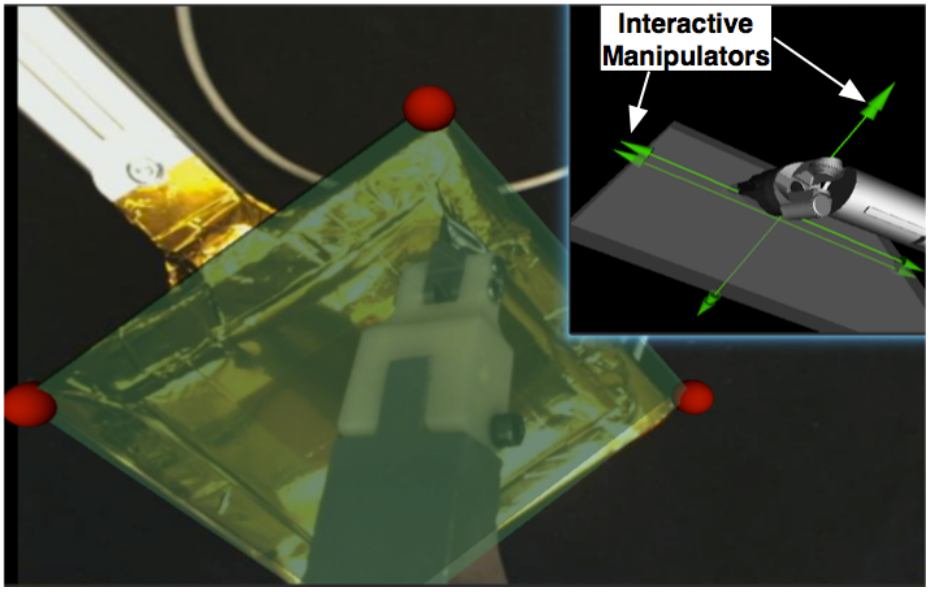


Figure 4.7: Virtual fixture modeling interface showing delayed video overlaid on top of registered model, and IMs for manipulating virtual fixtures in the modeling environment.

is not "back-drivable", we can compute the desired position/velocity based on user inputs, such as force measurement and joystick motion. For the telerobotics project, we are using a da Vinci MTM (master-tool-manipulator), which is an impedance type manipulator. Typically, impedance-type robots have low inertia and frictions, and are "back-drivable". We can compute the desired incremental motion by subtracting the "back-driven" actual robot end-effector position from the position that the robot low-level controller is command to servo to. For impedance-type robots, under quasi-static conditions, the user input force f_{user}, can be approximated by the incremental motion, or end-effector position/orientation errors, mimicking the force to incremental motion relationship on the admittance-type robot; this is also known as "pseudo-admittance" [3].

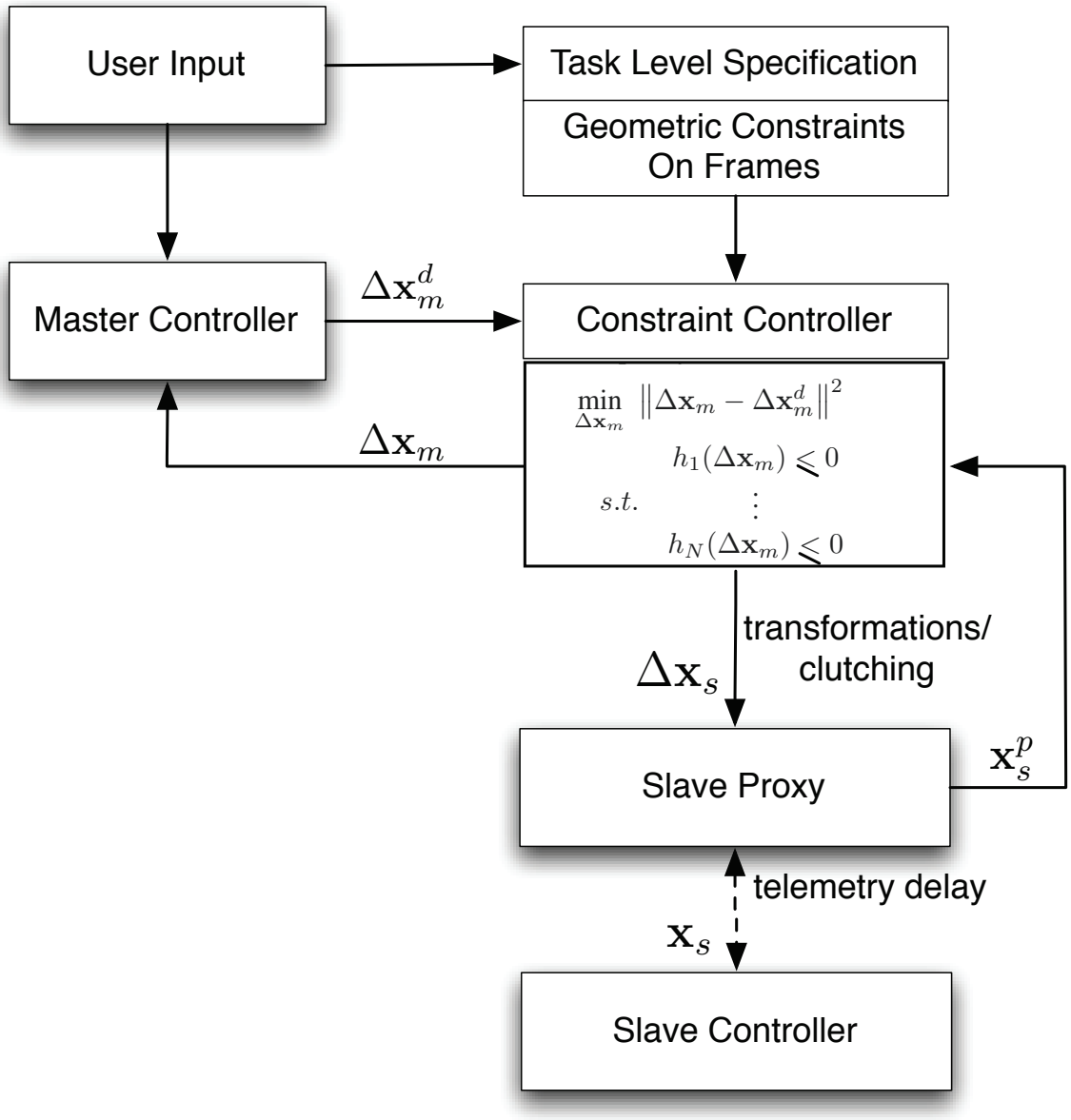


Figure 4.8: Master side control. $\Delta \mathbf{x}_m$ is the commanded (or optimal) master incremental motion, $\Delta \mathbf{x}_m^d$ is the desired master incremental motion, $\mathbf{x}_s^p$ is the slave proxy, and $\mathbf{x}_s$ is the actual slave position.

Joint level control

Incremental joint motion, $\dot{q}_m$ can be computed using inverse Jacobian methods. If the incremental Cartesian motions commanded are sufficiently small for each iteration, then, $\Delta X_m/\Delta t = J_m \Delta q_m/\Delta t$ approximates $\dot{X_m} = J_m \dot{q_m}$, where J_m is the manipulator Jacobian, and Δt is the period of the control loop. Ultimately, individual joints are servoed to commanded joint set points q_m by Proportional-Derivative (PD) controllers.

Control of Slave Proxy

We compute the commanded incremental Cartesian motion $\Delta \mathbf{x}_m$ in the quadratic constrained optimization problem in Equation (4.1). The commanded incremental Cartesian motion $\Delta \mathbf{x}_m$, is transformed into the slave coordinate system, $\Delta \mathbf{x}_s$ via an adjoint transformation.

The adjoint transformation matrix is a 6x6 matrix which transforms velocities (as twists) from one coordinate frame to another. Given the transformation frame T_m^s, the adjoint transformation $Ad_m^s : \mathbb{R}^6 \to \mathbb{R}^6$ is given as:

$$Ad_m^s = \begin{bmatrix} R & \hat{p}R \\ 0 & R \end{bmatrix} \tag{4.4}$$

$\Delta \mathbf{x}_s$ is integrated to the current slave proxy position to compute the next commanded slave proxy position, $\mathbf{x}_s^p$, which is then sent to the remote slave after the transmission time delay.

Clutching

During clutching, the slave proxy does not move, and is disengaged from the master. The master can translate, but not rotate relative to the simulated slave proxy to avoid becoming misaligned relative to each other and causing confusion for the operator.

4.3.5 Approximate Force Feedback

Studies have shown that force feedback significantly improves the sense of telepresence in teleoperation [17]. Due to the time delay between the operator station and the remote site, force feedback is instead simulated based on the interaction between the simulated slave and its environment in real-time.

The operator moves the master manipulator to generate a desired motion $\Delta\mathbf{x}_m^d$. The VF controller computes the optimal master incremental motion $\Delta\mathbf{x}_m$ subjected to virtual fixture constraints. The new master servo position $\mathbf{x}_m$ is computed through the integration of the $\Delta\mathbf{x}_m$ (Figure 4.8), and the PD controller servos each joint to this new position. As a result, the operator can feel the limits imposed by the virtual fixtures, and this haptic feedback serves as an approximation to the actual force derived from interaction with the environment.

4.3.6 Compliant Task Specification

The slave robot tracks the slave proxy's motion in free space. However, when the slave is in contact with its environment, it must compensate for the geometric errors between the environment model assumed at the master side and the actual environment to accurately execute the commanded motion without losing contact with the surface of the satellite, and without incurring excessive force.

Sources of errors stem from the models of the remote environment, such as a polyhedral approximations of the spacecraft, curvature of the tape, or the registration of this model with respect to the robot reference coordinate system. In the case of satellite servicing, we assume that a grapple procedure between the servicing spacecraft and the spacecraft being serviced has been performed, such that an initial registration between the robot and the CAD model of the satellite is provided.

Although geometric errors can lead to large variations of contact forces and moments, previous works have shown that an active (implicit) force controller can tolerate geometric uncertainties and apply a desired force to ensure proper contact, based on measured contact force [61]. We experimentally determined that it was necessary to apply a normal force toward the contact surface to prevent the Kapton tape from bunching up during the cutting procedure.

We transform the Task Model into a task frame that is used in a hybrid position/force controller on the slave robot [37, 57]. Hybrid position/force control allows us to control position along the free directions of motion, and regulate force con-

strained degrees of freedom in contact with the environment. The task specification of hybrid position/force control can be done through the use of *compliant task frame formalism* (TFF) [11, 49]. TFF provides an intuitive way to model and specify desired force and motion compatible with a compliance motion constraint. When the tool frame $\{t\}$ is in contact with the environment and is aligned with the compliant task frame $\{c\}$, the virtual fixture motion constraints for $\{t\}$ are compatible with the position and force controlled directions. Given á priori knowledge of the task and the virtual fixture primitives involved in $\{t\}$, we can determine the free directions in $\{c\}$ that are position controlled, and the restricted directions that should be force controlled, thereby providing online specifications to be executed by the hybrid position/force controller.

For example, the cutting task can be modeled as sliding a block on a frictionless surface (Fig. 4.9). In this task, force control regulates the force along the Z axis, and moments about the X and Y axes, respectively, to ensure contact and alignment of the contacting face of the block with the surface (restricted motion directions from the virtual fixture model). Tangential motion in the X and Y axes and rotation about the Z axis are position controlled and specified by the commanded incremental motion $\Delta\mathbf{x}_s$ (free motion directions from the virtual fixture model).

Given a particular task model (e.g., lines and/or planes), it is necessary to register this model to the remote environment in order to generate the correct task frame. Fig. 4.4 indicates that a *Coord. Transform* must be used to transform the task frame

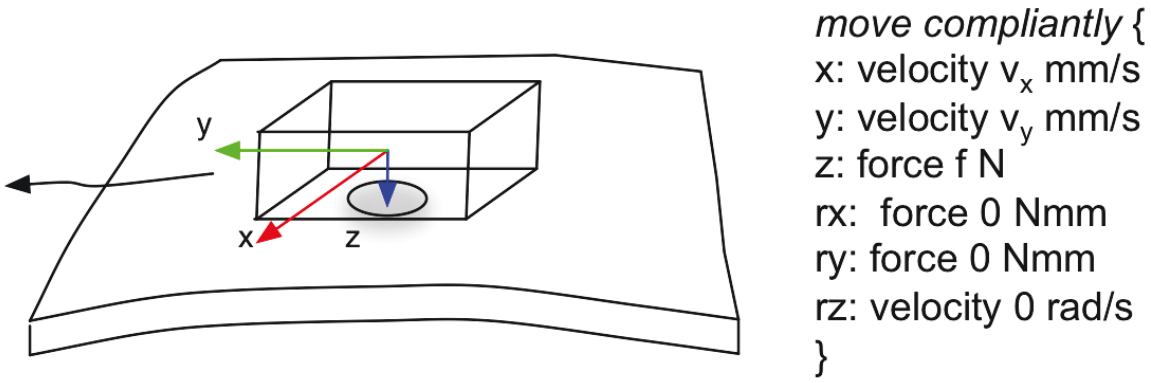


Figure 4.9: The cutting task can be modeled as sliding a block on a surface. Adapted from [11].

from the model coordinates to the physical coordinate system on the slave robot. The initial transformation can be obtained by detecting model features in calibrated stereo images or by teleoperating the slave robot to physically touch these features, but both of these methods are subject to errors.

For force-controlled directions, we can monitor the measured position of the robot and use that information to update the task frame [37]. For example, if the robot tool maintains contact with the surface, we can fit a plane to the history of measured robot positions and thereby compute the actual location of the plane. We can then update the transformation (*Update Transform*) to move the task frame so that the force control direction aligns with the actual normal to the plane.

Hybrid Position/Force Control

We implemented a standard Cartesian hybrid position/force controller for the slave robot. An outer force control loop is closed around an inner position controller,

where $\Delta f(t)$, the force tracking error between the desired force, $f_d(t)$, and the measured contact force, $f_c(t)$, is multiplied by a positive force gain, K_f to compute the desired velocity, $\dot{\mathbf{x}}_d(t)$. The velocity is integrated into a new position command for the slave robot [61]. This is the common approach for implementing force control for industrial robots where only Cartesian position commands are available.

$$\begin{aligned} \Delta f(t) &= f_c(t) - f_d(t) \\ \dot{\mathbf{x}}_d(t) &= -K_f \Delta f(t) \end{aligned} \tag{4.5}$$

We added saturation to limit the velocity in the range $[\dot{\mathbf{x}}_{min}, \dot{\mathbf{x}}_{max}]$ to avoid high impact velocities [37].

4.3.7 Task Monitoring

Isha Kandaswamy developed a task model for the blanket cutting task that enables the slave controller to quantify the expected sensor feedback [33]. For directions that are position-controlled, the measured force (using the JR3 force sensor) is compared to the expected force derived from the model. We performed several experiments to determine the model parameters for cutting tape with the two strategies (compression-based and tension-based). We found that a large component of the force in the direction of cutting is due to the friction between the cutter and tape surface, which could accurately be modeled as kinetic friction. We estimated the kinetic friction

by sliding the cutter along the tape (while not cutting). The force due to cutting the tape was adequately modeled as a constant, which we determined by measuring the force while cutting and then subtracting the previously measured frictional force. The final model is

$$|F_e| = \mu_k \cdot |F_n| + F_c, \tag{4.6}$$

where F_e is the expected force, F_n is the measured normal force, μ_k is the experimentally-determined coefficient of kinetic friction (0.56 and 0.48 for the compression-based and tension-based strategies, respectively), and F_c is the experimentally-determined cutting force (approximately 4N for both strategies). F_c and μ_k are the model parameters shown in Fig. 4.4.

If the *Task Monitor* detects a significant discrepancy between the measured and expected forces, the system stops the remote slave and alerts the operator. The operator can then try again, or use the augmented reality interface to examine the model and measured vs. expected results, and possibly modify the model parameters.

4.4 Implementation

4.4.1 Hardware System Components

We created a testbed for exploring telerobotic on-orbit servicing of spacecraft. The testbed consists of a modified da Vinci® robot system master console [23] (Figure

4.11(a)), and slave manipulators: a Whole Arm Manipulator (WAM) robot (Figure 4.11(b)) at Johns Hopkins University's Laboratory for Computational Sensing and Robotics and Motoman SIA10D robot (Figure 4.12) at the NASA Goddard Space Flight Center (GSFC) and the Research Technology Center at West Virginia University (WVU). These slave robots are used to emulate the remote on-orbit robots.

Operator's Master Console

The master console consists of Master-Tool-Manipulators (MTMs) from a da Vinci Classic Surgical system and a stereo display console from a da Vinci S Surgical System, both provided by Intuitive Surgical, Inc. The commercial version of the da Vinci is a clinically approved telerobotic system for Minimally Invasive Surgery (MIS). The MTM is a cable driven, fully actuated 7-DoF serial robot with two kinematic subsystems. A 3 DOF serial robot forms the first subsystem, similar to the PHANTOM device [50] and connects to the second subsystem, a redundant 4 DOF wrist mechanism. The operator positions his thumb and index finger on the non-actuated grippers. We refer the reader to [34] for details on the MTM. Figure 4.10 shows the closeup view of the da Vinci MTM wrist mechanism.

Through custom electronics hardware and software based on our real-time robotic control library, *cisst*, and the *Surgical Assistant Workstation* software framework [31, 32], we have created a modified system which gave us direct access to low-level joint control and we have implemented new features such as haptic feedback.

Remote Slave Systems

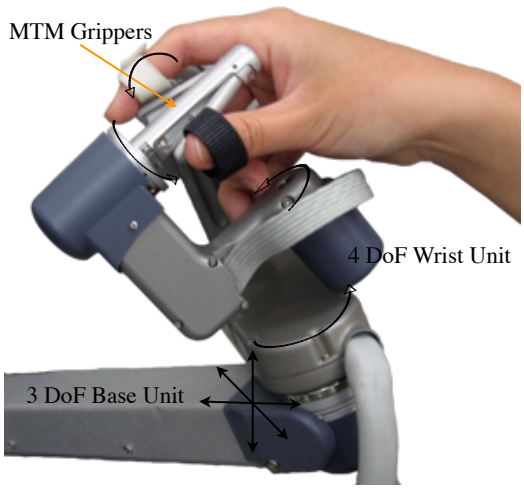


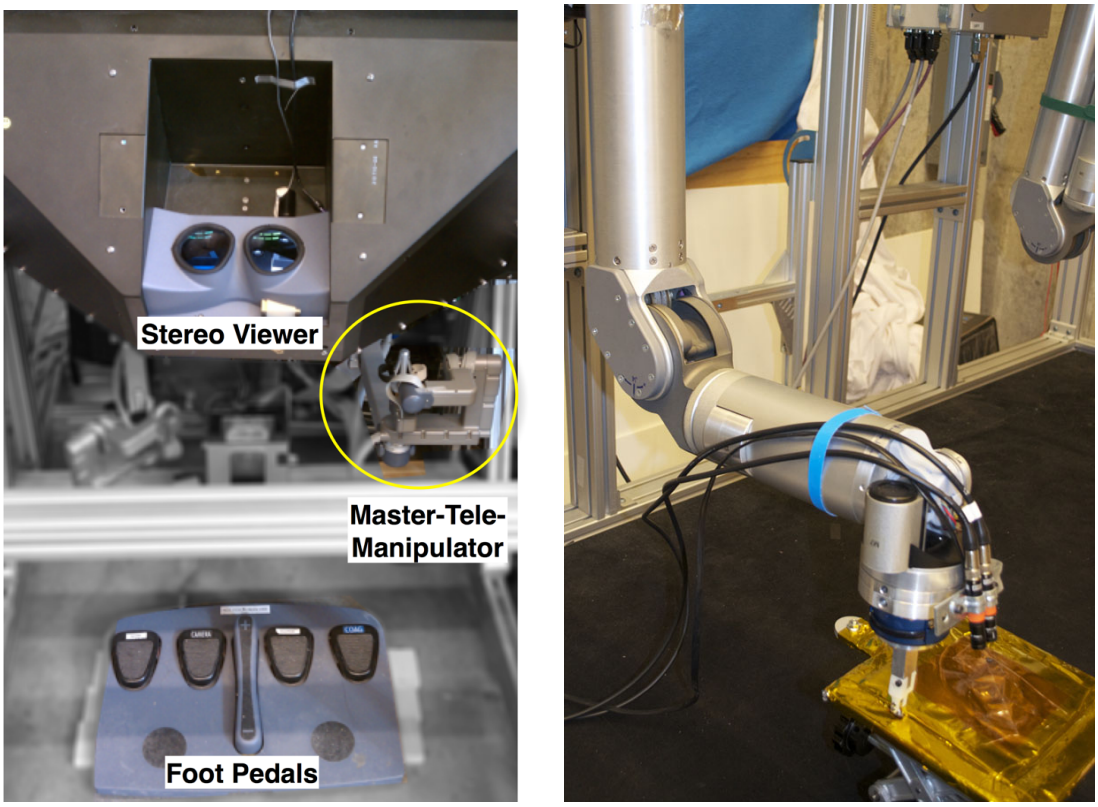
Figure 4.10: A closeup view of the da Vinci MTM wrist mechanism. Image credit: Ankur Kapoor [34].

The WAM™ Arm is a back-drivable serial manipulator from Barrett Technologies, Inc. We have the 7 DOF configuration, which has the 4 DOF WAM arm and the 3 DOF WAM wrist. While we use the WAM for our experiments at JHU, we occasionally use the Motoman SIA10D robot for project demonstration at either Goddard Space Flight Center or West Virginia University. The SIA10D series robot is a 7 DOF serial industrial robot with a payload capacity of 10 Kg.

A 6 DOF JR3® wrist force sensor is mounted at the end-effector of the WAM, and the blade of the cutter is mounted on the force sensor with one rapid-prototyped part designed to act as a "mechanical fuse" to avoid damaging the blade (Figure 4.13).

Foot Pedal Assembly

We integrated a da Vinci Surgical System foot pedal assembly as an input device into our system. The foot pedal has four pedals and a "+/-" rocker switch. We



(a) da Vinci Master Console (b) WAM slave with cutter and lipstick cameras

Figure 4.11: JHU teleoperation testbed for MLI cutting experiments.

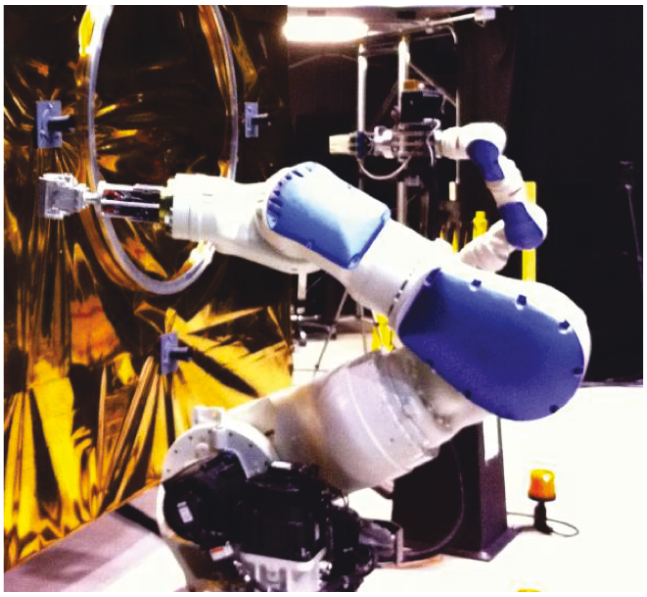


Figure 4.12: Motoman SIA10D.

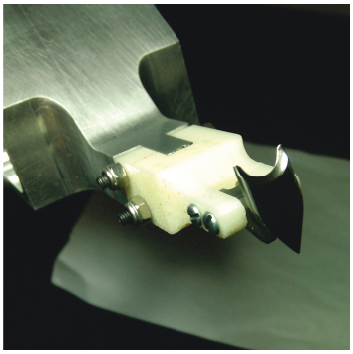


Figure 4.13: Cutter and rapid prototyped “mechanical fuse”.

Figure 4.14: The foot pedal assembly has four software configurable pedals and a "+/-" rocker switch.

use the "Camera" and "Clutch" pedals for getting operator input events during the registration step, or clutching operation, respectively.

Satellite Panel Mockup

The satellite surface is represented by a 20 × 20 cm aluminum plate that is covered with MLI. A small 15 × 15 cm MLI flap is taped on top of the plate to represent the fuel hatch (Fig. 4.2).

4.4.2 Software Components

Our control software relies heavily on the concept of component-based system design, where each component has well-defined interfaces for joint-level or Cartesian-level control, for example. As a result, we can quickly prototype new system configurations by connecting the master arm to any one of the remote slaves at run-time. The system component block diagram (Fig. 4.15) shows the interconnections between

different components of our system.

Delay Component

We developed a software delay component to simulate the time delay by circular-buffering the stereo video stream and telemetry data from the remote slave. The amount of time delay can be adjusted at run time. For our experiments, we configured the delay component for zero seconds (no added delay) or four seconds of time delay, as specified by our research collaborators at NASA GFSC and WVU.

Stereo Video Stream

The stereo video is obtained by two CS9001 Series Remote Head Color CCD cameras. The output is in NTSC format, with resolution of 748 (H) x 494 (V) pixels. Each camera requires an external camera controller that allows the adjustment of white balance, gamma, shutter, and other camera parameters. Each camera is 12 mm wide by 60 mm long (including lens) and can be mounted in a stereo vision setup with a baseline of only 20 mm. A short baseline is needed for the human stereoscopic vision to comfortably view closeup scenes.

We computed the intrinsic and extrinsic parameters of the stereo camera system using the Camera Calibration Toolbox for Matlab [1], in order to perform model overlay in the augmented reality environment.

The stereo video is streamed at a resolution of 600 x 400 pixels and at a refresh rate of less than 10 Hz. We did this for two reasons: one, we have network bandwidth limitations (between JHU and NASA GFSC, or between JHU and WVU) that prevent

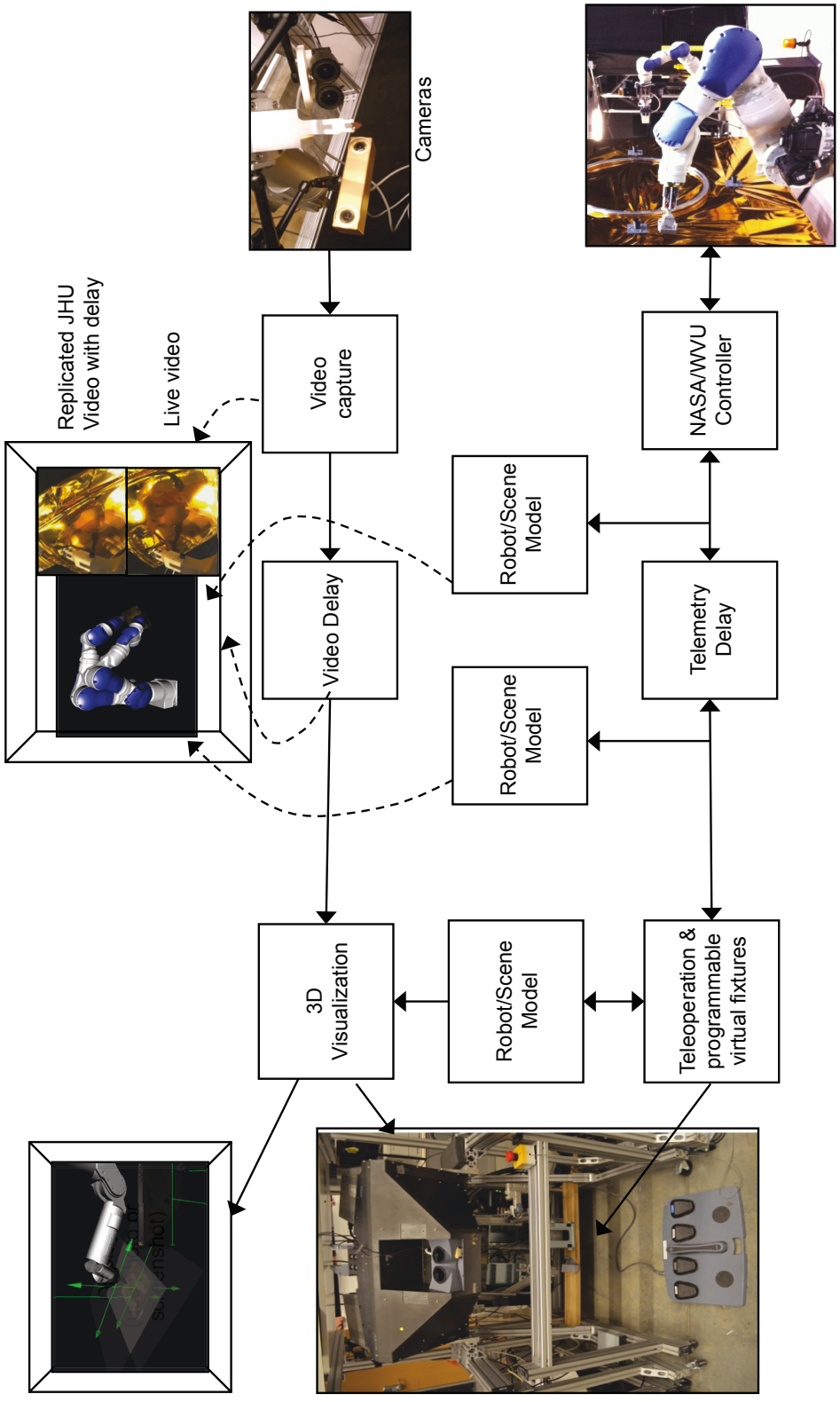


Figure 4.15: System block diagram

us from streaming at higher resolution and refresh rate; two, the video transmission quality for an actual space teleoperation mission is likely to be low and updated at low refresh rate as well.

Visualization and 3D User Interaction

OpenSceneGraph is an open source 3D graphics Application Programming Interface (API). It uses the OpenGL rendering pipeline and offers a scene graph data structure to represent the hierarchical geometric relationships between different nodes (e.g., the graphical models of robot links and satellite). We use OpenSceneGraph for visualization of the various graphical models of the robots, the cutting tool, satellite model, or virtual fixtures (Fig. 4.16).

We also use the osgManipulator NodeKit library to implement 3D user interaction for manipulating virtual fixture objects, which are also implemented as OpenSceneGraph nodes. The osgManipulator library offers mechanisms, called "draggers", to rotate, translate and scale the node that it is attached to. We attach several selectable draggers to a virtual fixture primitive node. The operator can then use an input device to drag or rotate the draggers, thereby applying rigid body transformations to the geometry associated with each virtual fixture. The associated geometric parameters, such as plane normal $\mathbf{n}$ and point $\mathbf{p}$ of each virtual fixture are used to generate motion constraints for teleoperation.

After registration, the virtual geometrical model of the satellite is shown in the coordinate system of the remote slave, and the operator can use the satellite model

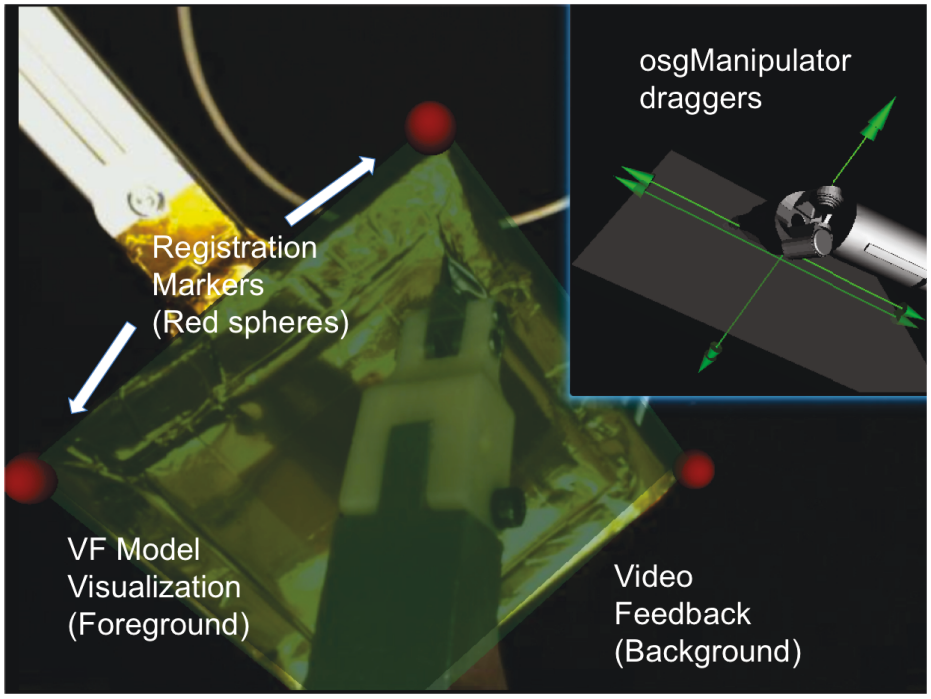


Figure 4.16: A "stay above plane" virtual fixture is overlaid on top of the MLI covered plate in the video feedback. osgManipulator draggers are manipulated to rotate and translate attached VF objects.

as a guide to interactively place plane or line virtual fixtures. If the stereo vision camera system on the remote slave is calibrated, we can overlay virtual fixtures, for example, a "stay above plane" virtual fixture, on top of the MLI covered plate in the video feedback (Fig. 4.16). In an actual satellite servicing mission, however, we expect that the autonomous satellite grappling procedure would produce a good initial registration between the satellite and the slave robotic system [2].

4.5 Experiment 1: Virtual Fixtures on Master Console

These experiments report the results of experiments performed at JHU using the da Vinci master console to teleoperate the WAM robot. The goal is to determine whether using virtual fixtures reduces operating time, and/or minimize damage to the MLI blanket. For these experiments, we accurately registered the slave robot (WAM) to the plate with the MLI blanket and did not use the hybrid position/force control on the slave side.

4.5.1 Experimental Setup

To evaluate the performance of our system we compare the results of the following experiments:

1. Unassisted teleoperation, with no time delay.

2. Unassisted teleoperation, with 4 second delay.

3. Virtual fixture-based teleoperation, with no time delay.

4. Virtual fixture-based teleoperation, with 4 second time delay.

Our virtual fixture-based teleoperation scenarios include a single line or a pair of planes.

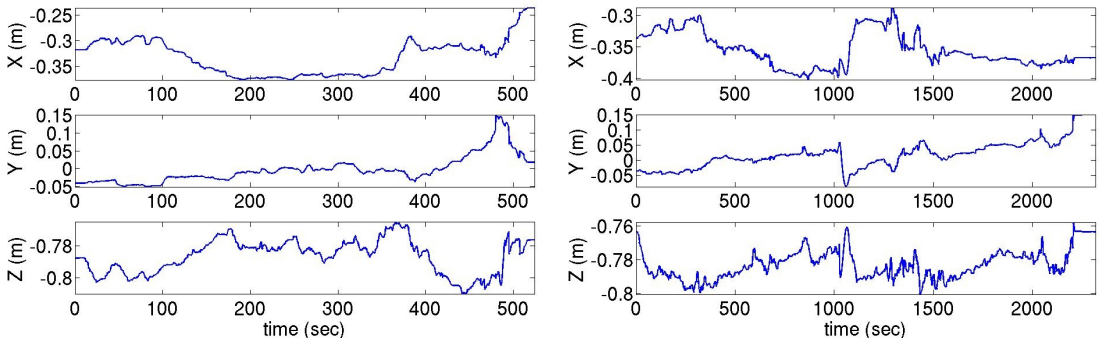
The task consists of cutting one side of the plate and comparing results based on the time of completion, effectiveness of the virtual fixture motion constraints, and damage done to the MLI. The plate was laid in the X-Y plane of the slave coordinate system and the task consisted of cutting the tape along the Y axis. For each experiment, the tool was moved close to a start corner on the plate to provide consistent initial conditions for the experiments. From there, the operator had to make an incision in the tape and slide the blade along the surface of the plate to cut the tape. For experiments using virtual fixtures, the operator had to manually adjust the position of each fixture in the virtual environment before starting the cut. The time for adjustments is not included in our results, as it is expected to be minimal with a well-implemented user interface.

4.5.2 Unassisted Teleoperation

In our first set of experiments, an operator was asked to cut one side of the tape around the MLI flap without any assistance. In the first experiment, no delay was injected in the system other than the unavoidable minimal (tens of milliseconds) delays caused by video frame grabbing, processing and local area network communications. The resulting trajectory of the experiment is shown in Fig. 4.17(a). In the second experiment, a four second delay was injected and the resulting trajectory is illustrated in Fig. 4.17(b).

A striking observation from the figures is the amount of time required to execute the task in the presence of time delay. While the task without time delay took approximately 8 minutes, the same task with time delay took over 30 minutes and resulted in severe damage to the MLI flap (Fig. 4.20(a)). In fact, most of the cutting during the experiments with time delay resulted in tearing the tape while pulling on the MLI flap.

We also fit a 3D line through the cutter's trajectory data points. The parameters of the line were estimated by total least squares and the direction of the line is reported in Table 4.3. While the expected direction of the cutting trajectory is $\mathbf{d} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T$ these results suggests that the direction of the trajectories deviate by 26° and 38°.



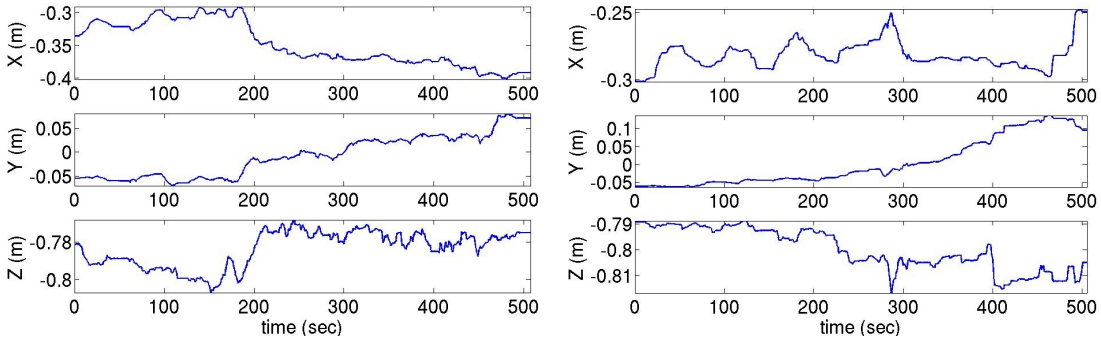
(a) No delay.

(b) 4 second delay.

Figure 4.17: Cutting with no virtual fixture.

4.5.3 Virtual Fixture Teleoperation: Planes

The second set of experiments considered using two planar virtual fixtures to constrain the position along the Z axis between both planes. Before executing the task, the operator adjusted the height of the planes to squeeze the tool between the lower and upper planes. To make the initial incision in the tape, the height of the lower plane must allow the tool to push on the surface to avoid gliding over the tape with the cutter. We repeated the experiment of section 4.5.2 with these fixtures and present the resulting trajectories in Fig. 4.18. The first observation is that with no time delay, the plane virtual fixtures only reduced the task completion time by 30 seconds. The direction of the best fit line (Table 4.3) also improved by deviating from the ideal line by dropping to 23° instead of 28°. The main improvement, however, occurs in the time delay scenario, where the unaided time of 33 minutes was reduced to less than 9 minutes by the addition of the virtual fixture. Furthermore, the direction



(a) No delay.

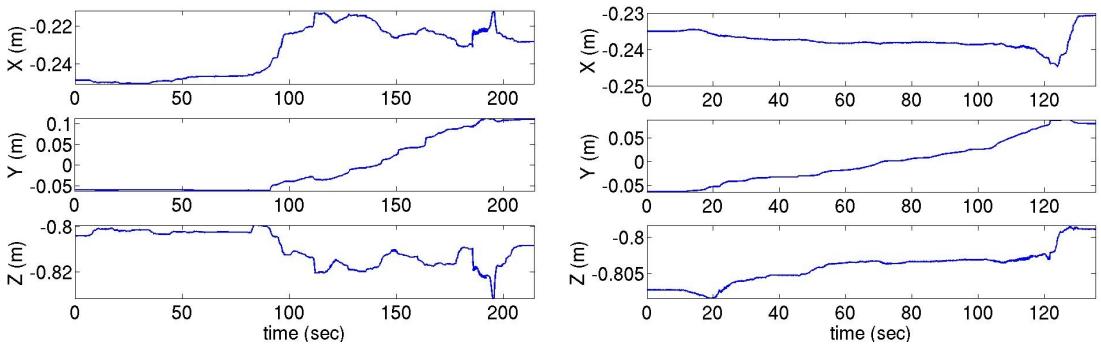
(b) 4 seconds delay.

Figure 4.18: Cutting with two planar virtual fixtures.

of the line that best fits the trajectory deviates from the ideal trajectory by only 6° instead of 38°. Finally, and significantly, there were no tears of the MLI when using the virtual planes. Our results indicate that when using virtual fixtures, the completion time is essentially independent of the time delay (at least for the time delays we tested).

4.5.4 Virtual Fixture Teleoperation: Line

In our last experiment, we replaced the two virtual planes with a virtual line. As with the other experiments, the cutter was placed near the flap where the virtual line was adjusted and activated. The resulting trajectories presented in Fig. 4.19 illustrate that the task, irrespective of time delays, was accomplished in less than 4 minutes which represents at 57% improvement over the experiments in 4.5.3 and, when time delays are injected, a 90% improvement. As presented in Table 4.3, the virtual lines



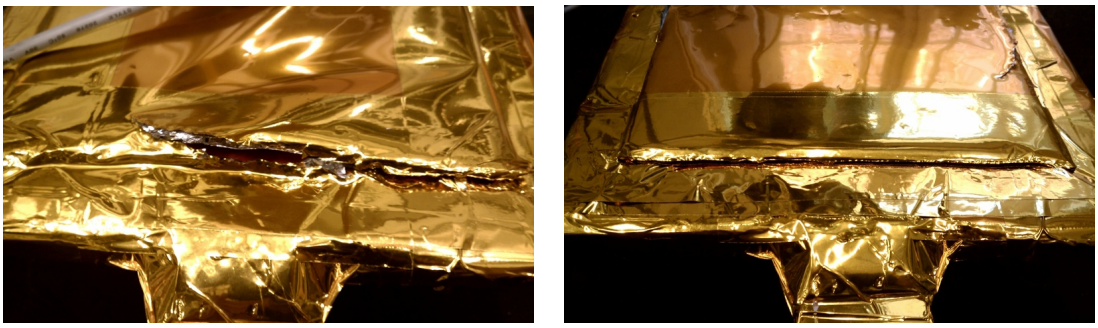
(a) No delay.

(b) 4 seconds delay.

Figure 4.19: Cutting with one line virtual fixture.

also improved the accuracy of the trajectory. The lines that best fit the trajectories only deviate from the ideal line by 8° and 2°. We also note that the MLI was left intact, as illustrated in Fig. 4.20(b).

Through these experiments, we have quantified the performance degradation due to time delay for the task in terms of time for task completion and trajectory deviation. We have also shown that virtual fixtures can effectively ameliorate the degradation in operator performance in time delayed teleoperation. We have also explored how the virtual fixture modeling environment allows the operator to progressively refine the virtual fixture motion constraints. The limitation is that each experiment setup is performed by one user, for one trial only.



(a) No virtual fixture.

(b) With line virtual fixture.

Figure 4.20: Cut results with 4 second delay.

Table 4.3: Estimated direction of cutting trajectory

	$\mathbf{d}^T$	$\arccos(\mathbf{d}^T \cdot \mathbf{y})$
Unassisted (0 sec)	$\begin{bmatrix} 0.4337 & 0.8978 & -0.0759 \end{bmatrix}$	26.1296
Unassisted (4 sec)	$\begin{bmatrix} -0.6061 & 0.7829 & 0.1402 \end{bmatrix}$	38.4731
Planes (0 sec)	$\begin{bmatrix} -0.3828 & 0.9179 & 0.1045 \end{bmatrix}$	23.3790
Planes (4 sec)	$\begin{bmatrix} -0.0077 & 0.9945 & -0.1048 \end{bmatrix}$	6.0120
Line (0 sec)	$\begin{bmatrix} 0.1263 & 0.9892 & -0.0743 \end{bmatrix}$	8.4283
Line (4 sec)	$\begin{bmatrix} -0.0180 & 0.9988 & 0.0448 \end{bmatrix}$	2.8072

4.6 Experiment 2: Virtual Fixtures on Master and Hybrid Position/Force Control on Slave

The experiment in Section 4.5 has shown that the use of virtual fixtures for teleoperation with time delay can significantly reduce the time required to complete the task and eliminate adverse events, such as tearing of the MLI, assuming no task geometric uncertainties. In practice, these uncompensated task geometric uncertainties can lead to large contact forces and moments.

This experiment focuses on compensating for these geometric uncertainties through the implementation of a hybrid position/force controller on the slave robot. The experimental setup is similar to the experiment described in Section 4.5. The plate is fixed so that it lies in the robot XY plane. The task model is represented on the master as a "stay above a plane" virtual fixture, i.e., planar surface of the satellite, and on the slave by a hybrid position/force controller, where force control is applied in the direction normal to the plane (Z) and position control is applied in the two directions tangent to the plane (X, Y). We are using the compression-based cutting strategy to cut the tape that secures the MLI patch (Eq. 4.6). For the force controlled direction, we fixed the desired force at -4 Newtons because our earlier experiments indicated that this force level provides good cutting performance [33]. An alternative

strategy would be to set the desired force based on the force applied by the operator (e.g., while pushing against the virtual fixture).

The tool orientation is fixed based on the plane normal (i.e., the planar surface of the tool is aligned with the plane defined in the task model). Other implementations could also be considered, such as defining a soft virtual fixture that would bias the tool to be perpendicular to the plane, while still allowing the operator to adjust the orientation.

4.6.1 Compensating for Orientation Error

In order to validate the controller's ability to compensate for task geometry errors, we performed several experiments where we deliberately introduced an orientation error of 15 degrees in the registration between the task frame and the slave robot coordinate system. The effect was that while the plate remained in the slave robot's XY plane, the commanded positions from the master were on a plane inclined at 15 degrees. The tool tip was also reoriented such that the tool tip was parallel to the cutting surface.

Fig. 4.21 shows the commanded and actual Z positions. Note that the commanded position is inclined, as expected, but the hybrid position/force controller adapts to the actual plate position, which is flat. The arc-tangent of the best-fit line to the commanded positions is -15 degrees, confirming the magnitude of the intentional misalignment. Fig. 4.22 shows that the slave robot force controller acts to maintain

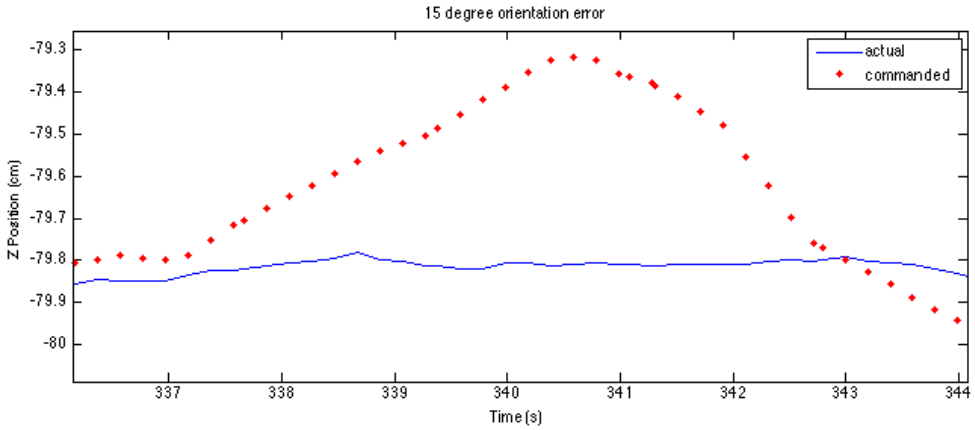


Figure 4.21: Tape cutting experiment with 15 degree orientation error of virtual fixture plane.

the commanded force of -4 Newtons in the direction normal to the plane.

4.7 Chapter Summary

Our initial task is to make a precise incision in the tape that fastens a flap of MLI blanket on the satellite, using a tool that bears some resemblance to a scalpel. We further note that most satellites currently in orbit have not been designed for servicing, much as the human body has not been designed to facilitate surgery.

Our testbed consists of a modified da Vinci master console, which we use to teleoperate various robots at JHU, NASA Goddard, and WVU. Compared to a typical surgical scenario, teleoperation of robots in space introduces significant time delays and bandwidth constraints.

We implemented software components to simulate time delays of telemetry and

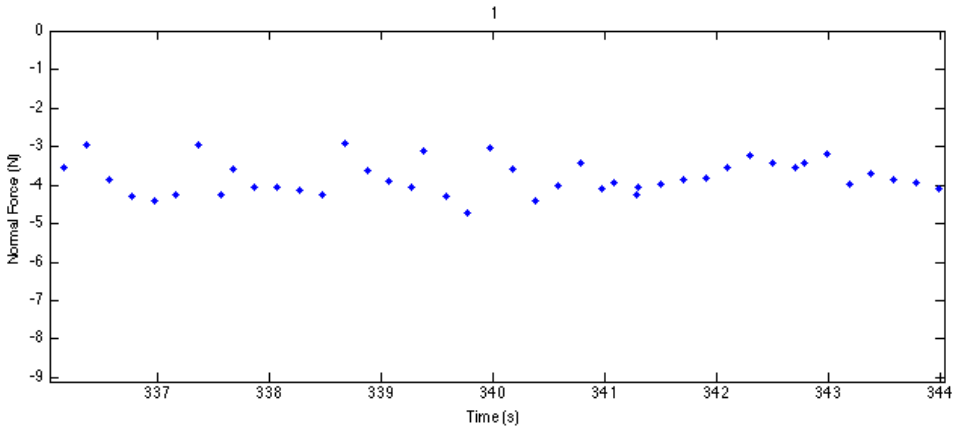


Figure 4.22: Tape cutting experiment with 15 degree orientation error of virtual fixture plane.

video. We do not yet consider bandwidth constraints, which may reduce the quality (resolution) of video feedback unless they can be overcome by image compression methods or future improvements to the communication infrastructure.

We have conducted a pilot study by teleoperating the WAM robot for an MLI cutting task using virtual fixtures with and without time delay. The results show that virtual fixtures reduce the time required to complete the task while also eliminating significant adverse events, such as tearing the MLI. The improvement in performance is especially dramatic when a simulated time delay (4 seconds) is introduced.

We also incorporated force sensor feedback at the remote site, where the slave robot executes the time delayed commanded motion consistent with the virtual fixture constraints by using a local, high-bandwidth hybrid force/position control scheme. Experimental results were presented to demonstrate the applicability of the controller to adapt to the actual task environment when the virtual fixture models were

subjected to geometric uncertainties.

4.8 Contributions

A spectrum of telerobotic techniques have been developed to cope with communication delay (2.2.2). We developed an augmented reality framework that enables the operator to design and implement assistive fixtures when confronted with a task in an uncertain environment unsuitable for autonomous systems.

This differs from prior telerobotic architectures. While the teleprogramming approach is limited by the difficulty of automatically understanding the operator's intent, we provide a system where the human operator can more explicitly program his/her intent. While the model-mediated approach attempts to extract the task model from the remote slaves sensor feedback and then sends this model back to the master, we use the sensor feedback to adjust the motion of the slave to better match the model on the master, rather than to update the model on the master.

Furthermore, our novel control approach combines concepts such as virtual fixtures, hybrid position/force control, task frame formalism and environment modeling, and is robust against modeling and registration errors in an uncertain environment. In addition, we are able to incorporate force sensor feedback at the remote site, where the slave robot executes the time delayed commanded motion consistent with the virtual fixture constraints by using a local, high-bandwidth hybrid force/position control

scheme.

We have created a framework for understanding and solving difficult problems in time-delayed operation, and built a satellite servicing test system to demonstrate the effective of our approach.

Chapter 5

Conclusion and Future Work

This dissertation focuses on the development of a system architecture for human-robot collaboration. This architecture is used to provide the human operator with sufficient and relevant information about the task and the task environment, through visual and force feedback. These feedbacks help the operator in determining high-level task goals and recognizing task constraints, thereby defining the virtual fixture task model through an interactive, or immersive, augmented reality interface. During task execution, the robot actively assist the human operator based on the virtual fixtures.

Although robotic neurosurgery and satellite servicing are in different application domains and have different requirements, we show that the system architecture, which emphasizes human-robot interaction in task model creation and robotic assisted execution can significantly improve overall task performance in each of the respective

tasks. This chapter summarizes the studies presented in this dissertation and addresses possible future directions of work.

5.1 Skull-base Surgery Application

We developed a cooperative control implementation of the system architecture for a skull-base surgery application. The motivation for introducing robotic assistance is to improve safety by preventing the surgeon from accidentally damaging critical neurovascular structures during the drilling procedure. The surgeon uses a graphical interface to segment regions of interest from the pre-operative CT images of the patient and simplifies the surface model to generate the virtual fixture task model. The surgical robotic system consists of a navigation system, a robotic arm with force sensor and visualization software and is used in a navigated, cooperatively-controlled fashion by the surgeon. During the execution, the virtual fixture is used to to constrain the motion of the robot-held cutting tool, so that it remains in the safe region that was defined on a preoperative CT scan. We conducted experiments on both foam skull and cadaver heads and concluded that the robot-assisted procedure is clinically feasible and provides some ergonomic benefits, such as stabilizing the drill. In addition, comparing the post-operative CT scans to the pre-operative CT scans show that the overcut (bone cut outside the virtual fixture boundary) can be as large as 3 mm and must be improved for clinical application.

Future Work

The effectiveness of the virtual fixture model depends on the accurate placement of the model in the robot workspace. The virtual fixture is defined in the coordinate system of the pre-operative image, and is typically registered to the intra-operative coordinate system, e.g., the robot, using an optical navigation system.

Some parts of the system architecture 2.6(b) remain to be implemented, such as the ability to compensate for registration error and/or intraoperative uncertainty. It is possible to characterize the amount of uncertainty in the registration and thereby decrease the virtual fixture margin to avoid overcut. It is possible to use intra-operative sensing, such as photo-acoustic ultrasound, to locate critical anatomic structures and update the virtual fixture model during the procedure.

5.2 Satellite Servicing Application

We implemented a telerobotics application of our proposed system architecture for satellite servicing tasks. Our goal is to perform robotic on-orbit servicing under ground-based teleoperation by human operators to perform tasks in the presence of uncertainty and time delay of several seconds. We presented an augmented reality interface, implemented on the master console of a modified da Vinci® surgical robot, that enables the operator to design and implement assistive virtual fixtures during teleoperation. Similar to the teleprogramming approach, the operator is essentially

interacting with a virtual world (a graphical simulation of the remote environment) with a virtual fixture model overlay. The virtual fixtures give immediate visual feedback and motion guidance to the operator, while the remote slave executes time delayed commanded motion using a local, high bandwidth hybrid position/force control scheme, which is robust against environment modeling and registration errors.

We conducted validation experiments by teleoperating a remote slave robot for a thermal barrier blanket cutting task using virtual fixtures with and without time delay. The results show that virtual fixtures reduce the time required to complete the task while also eliminating significant manipulation errors, such as tearing the blanket. The improvement in performance is especially dramatic when a simulated time delay (4 seconds) is introduced. By incorporating force sensor feedback at the remote site, where the slave robot executes the time delayed commanded motion consistent with the virtual fixture constraints by using a local, high-bandwidth hybrid force/position control scheme, we are also able to compensate for task geometric uncertainties that can lead to large contact forces and moments.

Future work

Registration Update and Task Monitor

The current slave side controller uses hybrid position/force control to compensate for geometric error, rather than directly updating registration. For our future work, we plan to implement the registration update on the slave side, based on force sensor

feedback. In addition, as the slave controller encounters various environmental features during task execution, it can record actual position and orientation and relay this data to the operator's station. The operator can use this information to refine the geometric relationships in the model, i.e., registration updates. A key issue in this process is to ensure consistent propagation of registration updates at the local and remote models.

Kandaswamy *et al.* [33] developed a task monitor for detecting cutting anomalies, such as bunching of the tape, that cause large increases in the measured force model, based on the expected sensor feedback. In Section 4.6, we collected force data to apply the task monitor offline, but did not enable the task monitor during the experiment. We plan to collect more cutting force data to validate if the model parameters (e.g., friction constant) that we determined through experiments in the lab (i.e., on earth) will be valid for MLI that we encounter in space, and also enable the task monitor in real-time, in order to detect and report cutting anomalies to the operator.

Additional Satellite Servicing Tasks:

NASA's Robotic Refueling Mission (RRM) outlines several other servicing tasks; among these, we are interested in the task of removal and installation of rotary fasteners for the insertion and removal of orbital replacement units (ORU). When replacing a defective ORU, several bolts must be unscrewed before the "drawer" like ORU module can be pulled out [2]. The bolt unscrewing task can be defined as rotating a tool, e.g., a screwdriver, along the axis of the bolt's rotation axis, while

keeping the tool in contact with the bolt. This can be defined with a combination of the "rotating about a line" and the "move along a line" virtual fixture primitives, by the operator, via the augmented reality interface.

Online Environment Model Refinement:

The model-mediated approach [54] attempts to build and update a simplified model of the unknown remote environment, as the slave robot interacts with the remote environment over time. We currently are able to use the force sensor to detect contact geometry and update the geometric relationships in the model through registration update. We can also use non-contact sensors, such as a laser 3D range sensor to detect objects of interest on the surface of the satellite, such as bolts on a ORU.

Bibliography

[1] Camera calibration toolbox for matlab.

[2] On-orbit satellite servicing study project report. Technical report, NASA Goddard Space Flight Center, http://ssco.gsfc.nasa.gov/servicing_study.html, 2010.

[3] J. J. Abbott and A. M. Okamura. Analysis of virtual fixture contact stability for telemanipulation. In *Proc. of IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2699–2706, 2003.

[4] J. J. Abbott and A. M. Okamura. Virtual fixture architectures for telemanipulations. In *Proc. of IEEE Int. Conf. on Robotics and Automation*, pages 2798–2806, Taipei, Taiwan, Sep 2003.

[5] B. A. Aikenhead, R. G. Daniell, and F. M. Davis. Canadarm and the space shuttle. *Journal of Vacuum Science Technology A: Vacuum, Surfaces, and Films*, 1(2):126–132, 1983.

[6] R. J. Anderson and M. W. Spong. Asymptotic stability for force reflecting

teleoperators with time delay. *Int. Journal of Robotics Research*, 11(2):135–149, 1992.

[7] K. Arun, T. Huang, and S. Blostein. Least-squares fitting of two 3-D point sets. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(5):698–700, Sep 1987.

[8] M. Balicki, A. Uneri, I. Iordachita, J. Handa, P. Gehlbach, and R. Taylor. Micro-force sensing in robot assisted membrane peeling for vitreoretinal surgery. In *Proc. of Medical Image Computing and Computer-Assisted Intervention*, pages 303–310. Springer, Sep 2010.

[9] A. K. Bejczy, W. S. Kim, and S. C. Venema. The phantom robot: predictive displays for teleoperation with time delay. In *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 546–551. IEEE, 1990.

[10] A. Bettini, P. Marayong, S. Lang, A. Okamura, and G. D. Hager. Vision-Assisted Control for Manipulation Using Virtual Fixtures. *IEEE Trans. on Robotics and Automation*, 20(6):953–966, 2004.

[11] H. Bruyninckx and J. De Schutter. Specification of force-controlled actions in the task frame formalism - a synthesis. *IEEE Trans. on Robotics and Automation*, 12(4):581–589, Aug 1996.

[12] K. Bumm, J. Wurm, J. Rachinger, T. Dannenmann, C. Bohr, R. Fahlbusch,

H. Iro, and C. Nimsky. An automated robotic approach with redundant navigation for minimal invasive extended transsphenoidal skull base surgery. *Minimally Invasive Neurosurgery*, 48(3):159–164, Jun 2005.

[13] F. Chan, I. Kassim, C. Lo, C. Ho, B. Ang, and I. Ng. Image guided robotic neurosurgery - in-vivo computer guided craniectomy. In *3rd Asian Conf. on Computer Aided Surgery (ACCAS)*, 2007.

[14] C.-C. Cheng. *Predictor displays–theory development and application to towed submersibles*. PhD thesis, Massachusetts Institute of Technology, 1991.

[15] B. L. Davies, S. J. Harris, W. J. Lin, R. D. Hibberd, R. Middleton, and J. C. Cobb. Active compliance in robotic surgery. *Surgery*, 211:285–292, 2002.

[16] P. Federspil, U. Geisthoff, D. Henrich, and P. Plinkert. Development of the first force-controlled robot for otoneurosurgery. *The Laryngoscope*, 113(3):465–471, 2003.

[17] W. Ferrell. Delayed force feedback. *Human Factors*, 8(5):449–455, 1966.

[18] J. Funda, T. Lindsay, and R. P. Paul. Teleprogramming: Toward delay invariant remote manipulation. *Presence*, 1(1):29–44, 1992.

[19] J. Funda, R. Taylor, B. Eldridge, S. Gomory, and K. Gruben. Constrained Cartesian motion control for teleoperated surgical robots. *IEEE Trans. on Robotics and Automation*, 12(3):453–465, Jun 1996.

[20] D. Glauser, P. Flury, and C. Burckhardt. Mechanical concept of the neurosurgical robot minerva. *Robotica*, 11:567–567, 1993.

[21] R. Goertz. Mechanical master-slave manipulator. *Nucleonics*, 12(11):45–59, November 1954.

[22] C. Guo, T. Tarn, and A. K. Bejczy. Fusion of human and machine intelligence for telerobotic systems. In *Proceedings of IEEE Int. Conf. on Robotics and Automation (ICRA)*, volume 3, pages 3110 – 3115, 1995.

[23] G. S. Guthart and K. Salisbury. The Intuitive™ telesurgery system: overview and application. In *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 618–621, Apr 2000.

[24] T. Haidegger, T. Xia, and P. Kazanzides. Accuracy improvement of a neurosurgical robot system. In *Proc. of 2nd IEEE RAS EMBS Int. Conf. on Biomedical Robotics and Biomechatronics*, pages 836 –841, Oct 2008.

[25] D. Handini, M. Y. Teo, and C. Lo. System integration of NeuroBot: a skull-base surgical robotic system. In *Proc. of IEEE Conf. on Robotics, Automation and Mechatronics*, pages 43–48, Singapore, Dec 2004.

[26] K. Hashtrudi-Zaad and S. Salcudean. Analysis and evaluation of stability and performance robustness for teleoperation control architectures. In *Proc. of the*

IEEE Int. Conf. on Robotics and Automation, volume 4, pages 3107 –3113 vol.4, 2000.

[27] G. Hirzinger, B. Brunner, J. Dietrich, and J. Heindl. Sensor-based space robotics-ROTEX and its telerobotic features. *IEEE Trans. on Robotics and Automation*, 9(5):649–663, Oct 1993.

[28] I. Iordachita, A. Kapoor, B. Mitchell, P. Kazanzides, G. Hager, J. Handa, and R. H. Taylor. Steady-hand manipulator for retinal surgery. In *MICCAI Medical Robotics Workshop*, pages 66–73, 2006.

[29] M. Jakopec, F. R. y Baena, S. Harris, P. Gomes, J. Cobb, and B. Davies. The hands-on orthopaedic robot "Acrobot": Early clinical trials of total knee replacement surgery. *IEEE Trans. on Robotics and Automation*, 19(5):902–911, Oct 2003.

[30] L. Joskowicz, R. Shamir, M. Freiman, M. Shoham, E. Zehavi, F. Umansky, and Y. Shoshan. Image-guided system with miniature robot for precise positioning and targeting in keyhole neurosurgery. *Computer Aided Surgery*, 11(4):181–193, 2006.

[31] M. Jung, T. Xia, A. Deguet, R. Kumar, R. Taylor, and P. Kazanzides. A Surgical Assistant Workstation (SAW) Application for Teleoperated Surgical Robot System. In *Proc. of MICCAI Workshop on Systems and Architecture for Computer Assisted Interventions*, Sep 2009.

[32] M. Y. Jung, A. Deguet, and P. Kazanzides. A component-based architecture for flexible integration of robotic systems. In *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 6107–6112, Oct 2010.

[33] I. Kandaswamy. Development, analysis and modeling of forces for cutting satellite insulation in telerobotic servicing missions. Master's thesis, Dept. of Computer Science, The Johns Hopkins University, 2012.

[34] A. Kapoor. *Motion Constrained Control of Robots for Dexterous Surgical Tasks.* PhD thesis, Johns Hopkins University, September 2007.

[35] A. Kapoor, A. Deguet, and P. Kazanzides. Software components and frameworks for medical robot control. In *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 3813–3818, Orlando, FL, May 2006.

[36] A. Kapoor, M. Li, and R. Taylor. Constrained Control for Surgical Assistant Robots. In *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 231–236, 2006.

[37] P. Kazanzides, N. Bradley, and W. Wolovich. Dual-drive force/velocity control: implementation and experimental results. In *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, volume 1, pages 92–97, May 1989.

[38] P. Kazanzides, J. Chang, I. Iordachita, J. Li, C. Ling, and G. Fichtinger. De-

velopment of an image-guided robot for small animal research. *Computer Aided Surgery*, 12(6):357–365, Nov 2007.

[39] P. Kazanzides, B. Mittelstadt, J. Zuhars, and R. Taylor. Force Sensing and Control for a Surgical Robot. In *Proc. of Int. Conf. on Robotics and Automation (ICRA)*, number 612-617, 1992.

[40] D. Kragic, P. Marayong, M. Li, A. M. Okamura, and G. D. Hager. Human-machine collaborative systems for microsurgical applications. *The Int. Journal of Robotics Research*, 24(9):731–741, Sep 2005.

[41] R. Kumar, T. M. Goradia, A. C. Barnes, P. Jensen, L. L. Whitcomb, D. Stoianovici, L. M. Auer, and R. H. Taylor. Performance of robotic augmentation in microsurgery-scale motions. In *Proc. of Medical Image Computing and Computer-Assisted Intervention*, pages 1108–1115. Springer, 1999.

[42] Y. Kwoh, J. Hou, E. Jonckheere, and S. Hayati. A robot with improved absolute positioning accuracy for ct guided stereotactic brain surgery. *IEEE Trans. on Biomedical Engineering*, 35(2):153–160, 1988.

[43] C. A. Lawn and B. Hannaford. Performance testing of passive communication and control in teleoperation with time delay. In *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 776–783. IEEE, 1993.

[44] J. Li, E. Balogh, I. Iordachita, G. Fichtinger, and P. Kazanzides. Image-guided

robot system for small animal research. In *1st Intl. Conf. on Complex Medical Engineering (CME)*, pages 194–198, Takamatsu, Japan, May 2005.

[45] M. Li, M. Ishii, and R. H. Taylor. Spatial Motion Constraints Using Virtual Fixtures Generated by Anatomy. *IEEE Trans. on Robotics*, 23(1):4–19, 2007.

[46] Q. Li, L. Zamorano, A. Pandya, R. Perez, J. Gong, and F. Diaz. The application accuracy of the NeuroMate robot - a quantative comparison with frameless and frame-based surgical localization systems. *Computer Aided Surgery*, 7(2):90–98, 2002.

[47] H. C. Lin, K. Mills, P. Kazanzides, G. D. Hager, P. Marayong, A. M. Okamura, and R. Karam. Portability and applicability of virtual fixtures across medical and manufacturing tasks. In *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 225–230. IEEE, May 2006.

[48] T. Machinis, K. Fountas, and V. Dimopoulos. History of acoustic neurinoma surgery. *Neurosugery Focus*, 18(4):19–27, 2005.

[49] M. T. Mason. Compliance and force control for computer controlled manipulators. *IEEE Trans. on System, Man, and Cybernetics*, 11(6), Jun 1981.

[50] T. Massie and K. Salisbury. The phantom haptic interface: A device for probing virtual objects. In *Proc. of the ASME International Mechanical Engineering Congress and Exhibition*, pages 295–302, 1994.

[51] M. Matinfar, C. Baird, A. Batouli, R. Clatterbuck, and P. Kazanzides. Robot assisted skull base surgery. In *Proc. of IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 865–870, 2007.

[52] C. Maurer, J. Fitzpatrick, M. Wang, R. Galloway, R. Maciunas, and G. Allen. Registration of head volume images using implantable fiducial markers. *IEEE Trans. on Medical Imaging*, 16(4):447–462, Aug 1997.

[53] P. Mitra. *Model Mediation for Time Delayed Teleoperation*. PhD thesis, Stanford, Sep 2008.

[54] P. Mitra and G. Niemeyer. Model-mediated telemanipulation. *Int. Journal of Robotics Research*, 27(2):253–262, Feb 2008.

[55] G. Niemeyer and J.-J. E. Slotine. Telemanipulation with time delays. *Int. Journal of Robotics Research*, 23:873–890, 2004.

[56] M. Oda. Experiences and lessons learned from the ETS-VII robot satellite. In *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 914–919, Apr 2000.

[57] M. H. Raibert and J. J. Craig. Hybrid position/force control of manipulators. *Trans. of the ASME*, 103(2):126–134, Jun 1981.

[58] M. Roche, P. O'Loughlin, D. Kendoff, V. Musahl, and A. Pearle. Robotic arm–

assisted unicompartmental knee arthroplasty: Preoperative planning and surgical technique. *American Journal of Orthopedics*, 38(2):10–15, 2009.

[59] L. B. Rosenberg. The use of virtual fixtures to enhance operator performance in time delayed teleoperation. Technical Report AL/CR-TR-1994-0139, Wright-Patterson Air Force Base, 1993.

[60] L. B. Rosenberg. Virtual Fixtures: Perceptual Tools for Telerobotic Manipulation. In *Proc. of Virtual Reality Annual International Symposium*, pages 76–82, 1993.

[61] J. Roy and L. Whitcomb. Adaptive force control of position/velocity controlled robots: theory and experiment. *IEEE Trans. on Robotics and Automation*, 18(2):121–137, Apr 2002.

[62] R. Sawaya, M. Hammoud, and D. Schoppa. Neurosurgical outcomes in a modern series of 400 craniotomies for treatment of parenchymal tumors. *Neurosurgery*, 42(5):1044–1055, 1998.

[63] C. P. Sayers, R. P. Paul, L. L. Whitcomb, and D. R. Yoerger. Teleprogramming for subsea teleoperation using acoustic communication. *IEEE Journal of Oceanic Engineering*, 23(1):60–71, 1998.

[64] T. B. Sheridan. *Telerobotics, automation, and human supervisory control*. MIT Press, 1992.

[65] T. B. Sheridan. Space teleoperation through time delay: review and prognosis. *IEEE Trans. on Robotics and Automation*, 9(5):592–606, 1993.

[66] C. Sim, W. S. Ng, M. Y. Teo, Y.-C. Loh, and T. T. Yeo. Image-guided manipulator compliant surgical planning methodology for robotic skull-base surgery. In *Proc. of Int. Workshop on Medical Imaging and Augmented Reality*, pages 26–29, Hong Kong, China, Jun 2001.

[67] R. Taylor, P. Jensen, L. Whitcomb, A. Barnes, R. Kumar, D. Stoianovici, P. Gupta, Z. Wang, E. DeJuan, and L. Kavoussi. A steady-hand robotic system for microsurgical augmentation. *Int. Journal of Robotics Research*, 18(12):1201–1210, 1999.

[68] R. Taylor, H. Paul, B. Mittelstadt, W. Hanson, P. Kazanzides, J. Zuhars, E. Glassman, B. Musits, W. Williamson, and W. Bargar. An image-directed robotic system for precise orthopaedic surgery. *IEEE Trans. on Robotics and Automation*, 80(3):261–275, 1994.

[69] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. and Mach. Intell.*, 13(4):376–380, Apr 1991.

[70] T. Xia, C. Baird, G. Jallo, and K. Hayes. An integrated system for planning, navigation and robotic assistance for skull base surgery. *Int. Journal of Medical Robotics and Computed Assisted Surgery*, pages 321–330, Sep 2008.

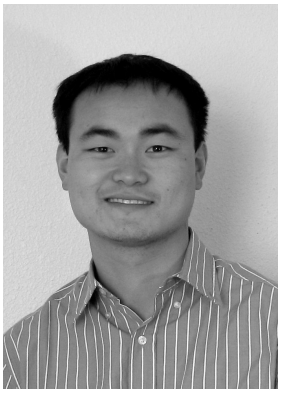
[71] T. Xia, A. Kapoor, P. Kazanzides, and R. Taylor. A constrained optimization approach to virtual fixtures for multi-robot collaborative teleoperation. In *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 639 –644, Sep 2011.

[72] T. Xia, S. Leonard, A. Deguet, L. L. Whitcomb, and P. Kazanzides. Augmented reality environment with virtual fixtures for robotic telemanipulation in space. In *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 5059–5064, Oct 2012.

[73] T. Xia, S. Leonard, I. Kandaswamy, A. Blank, L. L. Whitcomb, and P. Kazanzides. Model-based telerobotic control with virtual fixtures for satellite servicing tasks. In *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, May 2013.

[74] K. Yoshida. Achievements in space robotics. *IEEE Robotics Automation Magazine*, 16(4):20 –28, Dec 2009.

Vita



Tian Xia received the B.S. degree in Electrical Engineering from University of Washington in Seattle, Washington, and enrolled in the Computer Science Ph.D. program at the Johns Hopkins University. He received a National Science Foundation Graduate Research Fellowship in 2008.